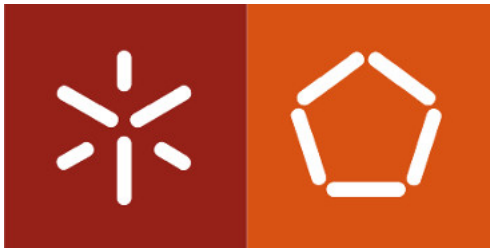

Universidade do Minho
Licenciatura em Engenharia Informática

Computação Gráfica

Fase 3 - Grupo 20

André Silva - A87958
Armando Silva - A87949
Joana Oliveira - A87956
João Nunes - A87972

Maio 2022




Conteúdo

Lista de Figuras

1 Introdução

Nesta terceira fase do projeto, é proposta a continuação do desenvolvimento do sistema solar num cenário gráfico 3D. Desta vez, foi necessária a implementação de curvas, superfícies cúbicas e VBOs.

Para o gerador, foi nos pedida a adição da possibilidade de criarmos um novo ficheiro 3d, este que precisa de um nome de ficheiro, que contém um conjunto de *control points* de Bezier, e também do nível de tecelagem.

Já para o motor, é necessária a implementação de transformações com tempo. A translação tem de ser realizada através da curva cúbica, Catmull-Rom, e a rotação tem de utilizar o tempo para puder efetuar uma rotação de 360 graus ao torno de um eixo. Ambas as transformações decorrerão num certo tempo dado. Tudo isto com o objetivo de criar animações com base nestas curvas. Ainda é nos pedida, a alteração do modo de desenho das primitivas para VBOs.

Neste relatório, apresentamos e explicamos todas as funcionalidades pedidas no enunciado divididas em várias secções.

2 Generator/Gerador

Em relação ao gerador foi adicionada a leitura de *patches* de Bezier, que estejam guardadas num ficheiro, bem como a sua transformação para pontos e a respetiva escrita num ficheiro .3d.

2.1 Leitura dos Bezier patches

A leitura dos ficheiros ".patch"é feita através da função **readBezier**. As variáveis *all_points* e *patchesIndices* são criadas e correspondem, respetivamente, a todos os control points e índices de todos os patches. O preenchimento das mesmas é feito através da leitura linha a linha do ficheiro. No fim da leitura é chamada a função **parseBezierPatches**, que escreve os pontos criados pelas curvas. A criação destes pontos será explicada no tópico seguinte.

2.2 Cálculo dos pontos

Como já foi anteriormente dito, a função **parseBezierPatches** cria diferentes matrizes para cada coordenada x, y ou z de todos os pontos. De seguida é aplicada a função **getBezierPoint**, em função da tecelagem, sendo que esta irá sendo incrementada sucessivamente. Esta função segue o seguinte algoritmo:

$$B(u,v) = \lfloor u^3 \quad u^2 \quad u \quad 1 \rfloor M \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} M^T \begin{bmatrix} v^3 \\ v^2 \\ v \\ 1 \end{bmatrix}$$

Figura 1. Algoritmo de Bezier

```
for (int s = 0; s < patchesIndices.size(); s+=16){

    for (size_t a = 0; a < 4; a++) {

        for (size_t b = 0; b < 4; b++) {
                matrixX[a][b] = all_points.at(patchesIndices.at(s + a * 4 + b))[0];
                matrixY[a][b] = all_points.at(patchesIndices.at(s + a * 4 + b))[1];
                matrixZ[a][b] = all_points.at(patchesIndices.at(s + a * 4 + b))[2];
        }
    }

    for (int i = 0; i < tesselation; i++){

        for (int j = 0; j < tesselation; j++){
            counter++;
            u = t * i;
            v = t * j;
            float next_u = t * (i+1);
            float next_v = t * (j+1);

            getBezierPoint(u, v, (float**)matrixX, (float**)matrixY, (float**)matrixZ, pos[0]);
            getBezierPoint(u, next_v, (float**)matrixX, (float**)matrixY, (float**)matrixZ, pos[1]);
            getBezierPoint(next_u, v, (float**)matrixX, (float**)matrixY, (float**)matrixZ, pos[2]);
            getBezierPoint(next_u, next_v, (float**)matrixX, (float**)matrixY, (float**)matrixZ, pos[3]);
```

Figura 2. Cálculo das matrizes presente na função **parseBezierPatches**

```
void getBezierPoint(float u, float v, float** matrixX, float** matrixY, float** matrixZ, float* pos){
    float bezierMatrix[4][4] = { { -1.0f, 3.0f, -3.0f, 1.0f },
                                 { 3.0f, -6.0f, 3.0f, 0.0f },
                                 { -3.0f, 3.0f, 0.0f, 0.0f },
                                 { 1.0f,  0.0f, 0.0f, 0.0f } };

    float vetorU[4] = { u * u * u, u * u, u, 1 };
    float vetorV[4] = { v * v * v, v * v, v, 1 };

    float vetorAux[4];
    float px[4];
    float py[4];
    float pz[4];

    float mx[4];
    float my[4];
    float mz[4];

    multMatrixVector((float*)bezierMatrix, vetorV, vetorAux);
    multMatrixVector((float*)matrixX, vetorAux, px);
    multMatrixVector((float*)matrixY, vetorAux, py);
    multMatrixVector((float*)matrixZ, vetorAux, pz);

    multMatrixVector((float*)bezierMatrix, px, mx);
    multMatrixVector((float*)bezierMatrix, py, my);
    multMatrixVector((float*)bezierMatrix, pz, mz);


    pos[0] = (vetorU[0] * mx[0]) + (vetorU[1] * mx[1]) + (vetorU[2] * mx[2]) + (vetorU[3] * mx[3]);
    pos[1] = (vetorU[0] * my[0]) + (vetorU[1] * my[1]) + (vetorU[2] * my[2]) + (vetorU[3] * my[3]);
    pos[2] = (vetorU[0] * mz[0]) + (vetorU[1] * mz[1]) + (vetorU[2] * mz[2]) + (vetorU[3] * mz[3]);

}
```

Figura 3. Cálculo dos pontos

3 Engine/Motor

Relativamente ao motor foi feita a atualização das transformações *translate* e *rotate*, que agora são feitas usando o tempo como variável. A implementação da curva cúbica Catmull-Rom foi necessária, isto para, em conjunto com as transformações, criar animações. Além disso, o modo de desenho das primitivas começou a ser feito com recurso a VBOs.

Durante o desenvolvimento desta fase deparámo-nos com um erro na forma como as transformações são feitas. Assim sendo, as transformações normais e, posteriormente, as dinâmicas, foram alteradas de forma a respeitar a ordem das transformações presentes no ficheiro XML, algo que não acontecia na fase anterior.

3.1 Leitura do Ficheiro XML

Tendo em conta as novas transformações implementadas e também o erro anteriormente referido, foi necessário alterar a forma como lemos o ficheiro XML. A função **parseGroup** é agora também capaz de ler as novas variáveis do ficheiro, i.e, o *time*, *align* e os novos pontos para a translação. Para isso foi criada uma nova classe *Transformation*, que guarda ordenamente cada transformação lida no ficheiro XML. Cada transformação inclui o nome do tipo de transformação, os seus pontos, ângulos, tempos e align, caso existam. Esta classe é guardada num vector pertencente à classe *Group*.

```
private:
    string name;
    std::vector<float> coordinates;
    float angle = -1;
    float time = -1;
    bool align;
```

Figura 4. Variáveis da classe *Transformation*

```
private:
    std::vector<std::string> file;
    std::vector<Transformation> transformation;
    std::vector<Group> subgroups;
    std::vector<std::vector<float>> total_figures;
```

Figura 5. Variáveis da classe *Group*

Após a leitura das transformações, vem a leitura dos "models". Estes são agora lidos diretamente da função **parseGroup**, em vez da **drawFigures**, de forma a otimizar o programa. A função **parse3D**, que realiza a leitura dos ficheiros .3d, também foi alterada e será explicada na secção dos VBOs.

3.2 Desenho das Figuras

A função **drawFigures** foi alterada de modo a aplicar estas novas transformações. Agora é percorrido o vetor das transformações dos grupos e estas são aplicadas pela ordem em que surgem. De forma a diferenciar as transformações normais das transformações dinâmicas é analisada a variável *time*. Caso esta seja -1 aplicam-se as transformações normais, já feitas na fase anterior. Caso contrário, é aplicada uma das novas transformações. Para a translação usamos como auxílio a função **renderCatmumllRomCurve**.

3.3 Transformações com Tempo

As transformações com tempo são aplicas nas translações e rotações. Para acompanhar a variação do tempo é utilizada a função gluGet(GLUT_ELAPSED_TIME), tal como foi recomendado.

Na rotação apenas é aplicado o cálculo apresentado na figura seguinte, que determina o ângulo a ser usado na função **glRotatef**. Além disso, foi também novamente inserida a funcionalidade existente na Fase 1, que foi posteriormente removida na Fase 2, e que permite alterar a forma como observamos as figuras, ou seja, alternar entre figuras a cheio ou a linhas. Isto ajuda-nos a perceber melhor o movimento de rotação dos planetas e do sol.

```
float angle = (((float)glutGet(GLUT_ELAPSED_TIME) / 1000) * 360) / (float)(transf.getTime());
glRotatef(angle, coords[0], coords[1], coords[2]);
```

Figura 6. Rotação

Já na translação são feitos mais cálculos, dependendo da variável *align* presente no ficheiro XML. Independemente do valor da variável referida, será sempre executada a função **renderCatmullRomCurve**. Esta função recebe os pontos de translação já armazenados do ficheiro XML e desenha a curva de Catmull. A possibilidade de adicionar e remover estas curvas foi adicionada como extra.

```
// Draws Curve
void renderCatmullRomCurve(std::vector<float> translate) {

// draw curve using line segments with GL_LINE_LOOP
    float pos[3], deriv[3];

    float gt = 0.00f;

    glBegin(GL_LINE_LOOP);
    while(gt < 1){
        getGlobalCatmullRomPoint(translate, gt, pos, deriv);
        glVertex3f(pos[0], pos[1], pos[2]);
        gt += 0.01f;
    }
    glEnd();
}
```

Figura 7. Função que desenha a curva

De seguida é chamada a função **getGlobalCatmullRomPoint** que nos dará os pontos de acordo com os quais a figura se deverá mover. Por fim, a variável *Align* é verificada e, caso esta seja verdadeira, são executadas as funções presentes na figura abaixo. Caso contrário, estas funções serão ignoradas.

As funções presentes na imagem abaixo fazem com que a figura esteja alinhada com curva.

```
if (transf.getAlign()){
    float x[3] = {deriv[0], deriv[1], deriv[2]};
    normalize(x);
    float y0[3] = {0,1,0};
    float z[3];
    cross(x,y0,z);
    normalize(z);
    float y[3];
    cross(z,x,y);
    normalize(y);

    float matrix[16];
    buildRotMatrix(x,y,z, matrix);

    glMultMatrixf(matrix);
}
```

Figura 8. Funções a ser executadas caso Align seja "True"

3.4 Implementação de VBOs

Para esta fase foi também necessário implementar o desenho das figuras utlizando VBOs. Esta implementação encontra-se nas funções **parse3D** e **drawFigures**.

Primeiramente é criado um vector global v que armazena todos os pontos de todas as figuras. Na *main* são chamadas as funções **glGenBuffers**, **glBindBuffer** e **glBufferData**, que efetuam a cópia do vector para a memória gráfica.

No **parse3D** é preenchido o vector v, durante a leitura do ficheiro, com todos os pontos do ficheiro .3d. No **drawFigures** efetuamos o desenho dos VBOs. Primeiramente é chamada a função **glBindBuffer** para ativar o VBO, de seguida a **glVertexPointer** para informar que cada vértice é formado por 3 floats e, por último, a função **glDrawArrays**, tendo em conta o número de vértices a desenhar para efetuar o desenho.

4 Sistema Solar

Em relação à versão do sistema solar da fase anterior foram agora adicionadas novas funcionalidades. O ficheiro XML foi alterado de modo a que os planetas tenham movimentos de rotação e de translação. Todos os planetas, exceto Mercúrio, juntamente com as luas fazem o seu movimento de translação em sentido anti-horário. Em relação ao movimento de rotação, todos os planetas, à exceção de Vénus e Urano, o fazem também no sentido anti-horário. Além disso foi também adicionado um cometa, recorrendo às curvas de Bezier, cujo movimento de translação é diferente de todos os planetas.

Nas figuras seguintes é possível observar o resultado obtido nesta fase relativamente ao nosso sistema solar.

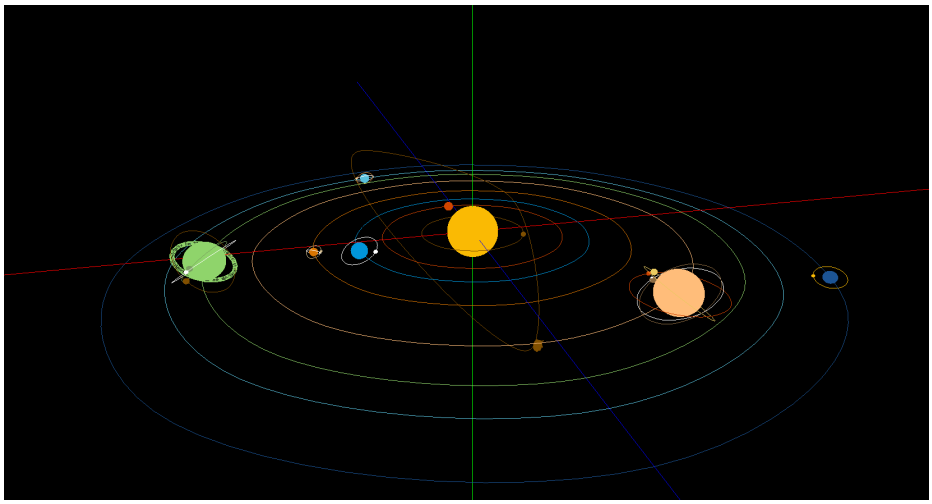

Figura 9. Versão do sistema solar com as rotas visíveis

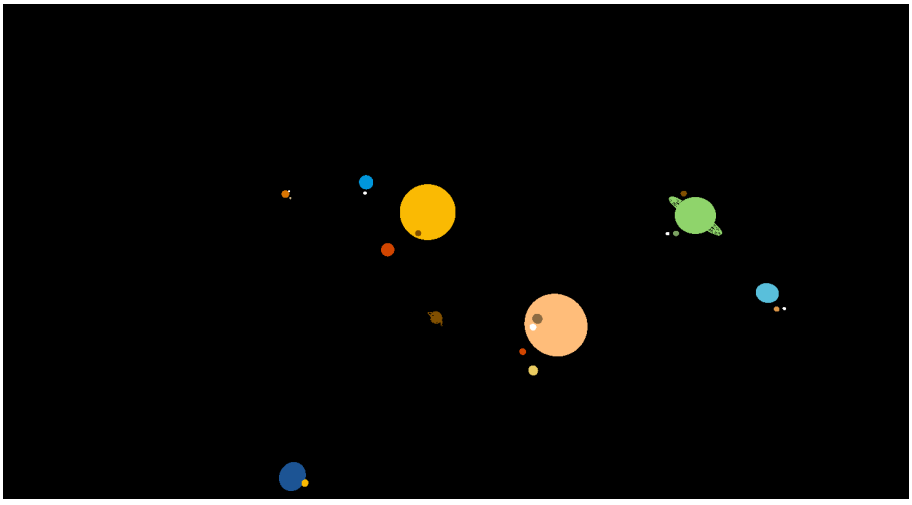

Figura 10. Versão do sistema solar sem as rotas visíveis

5 Conclusão

Dado por concluída a terceira fase deste projeto, consideramos importante realçar todos os pontos positivos e negativos, e ainda, efetuar uma análise crítica final do trabalho realizado.

A passagem para o desenho das primitivas através dos VBOs foi de fácil implementação. Outro ponto positivo, é a demo final construída que representa bem o sistema solar, tal como pedido. Também é de notar a perceção do erro encontrado relativamente às transformações.

Para concluir, esta fase cumpre com todos os requisitos propostos apesar de várias dificuldades sentidas durante todo o trabalho, e como tal, o grupo sente que este está positivo.

A Ficheiro XML

```
<world>
    <camera>
        <position x="10" y="10" z="10"/>
            <lookAt x="0" y="0" z="0"/>
            <up x="0" y="1" z="0"/>
        <projection fov="60" near="1" far="1000"/>
    </camera>
    <group>
        <!--SOL-->
        <transform>
            <color r="0.98" g="0.73" b="0.01"/>
            <rotate time="3" x="0" y="1" z="0" />
        </transform>
        <models>
            <model file="sphere.3d"/>
        </models>
    </group>
    <group>
        <!--MERCURIO-->
        <transform>
            <color r="0.5" g="0.31" b="0.0"/>
            <translate time = "3" align="True">
                <point x = "1.41421" y = "0" z = "1.41421" />
                <point x = "0" y = "0" z = "2" />
                <point x = "-1.41421" y = "0" z = "1.41421" />
                <point x = "-2" y = "0" z = "0" />
                <point x = "-1.41421" y = "0" z = "-1.41421" />
                <point x = "0" y = "0" z = "-2" />
                <point x = "1.41421" y = "0" z = "-1.41421" />
                <point x = "2" y = "0" z = "0" />
            </translate>
            <rotate time="5" x="0" y="1" z="0" />
            <rotate angle="-30.0" x="0.0" y="1.0" z="0.0"/>
            <scale x="0.1" y="0.1" z="0.1"/>
        </transform>
        <models>
            <model file="sphere.3d"/>
        </models>
    </group>
    <group>
        <!--VENUS-->
        <transform>
            <color r="0.82" g="0.27" b="0.0"/>
            <translate time = "7" align="True">
                <point x = "3.5" y = "0" z = "0" />
                <point x = "2.47487" y = "0" z = "-2.47487" />
                <point x = "0" y = "0" z = "-3.5" />
                <point x = "-2.47487" y = "0" z = "-2.47487" />
                <point x = "-3.5" y = "0" z = "0" />
                <point x = "-2.47487" y = "0" z = "2.47487" />
                <point x = "0" y = "0" z = "3.5" />
```

```
            <point x = "2.47487" y = "0" z = "2.47487" />
        </translate>
        <rotate time="7" x="0" y="-1" z="0" />
        <rotate angle="90" x="0" y="1" z="0"/>
        <scale x="0.2" y="0.2" z="0.2"/>
    </transform>
    <models>
        <model file="sphere.3d"/>
    </models>
</group>
<group>
    <!--TERRA-->
    <transform>
        <color r="0.0" g="0.59" b="0.86"/>
         <translate time = "10" align="True">
            <point x = "4.5" y = "0" z = "0" />
            <point x = "3.18198" y = "0" z = "-3.18198" />
            <point x = "0" y = "0" z = "-4.5" />
            <point x = "-3.18198" y = "0" z = "-3.18198" />
            <point x = "-4.5" y = "0" z = "0" />
            <point x = "-3.18198" y = "0" z = "3.18198" />
            <point x = "0" y = "0" z = "4.5" />
            <point x = "3.18198" y = "0" z = "3.18198" />
        </translate>
        <rotate time="3" x="0" y="1" z="0" />
        <rotate angle="233" x="0" y="1" z="0"/>
        <scale x="0.3" y="0.3" z="0.3"/>
    </transform>
    <models>
        <model file="sphere.3d"/>
    </models>
    <group>
        <!--LUA-->
        <transform>
            <color r="1.0" g="1.0" b="1.0"/>
            <rotate angle="26.57" x="1" y="0.0" z="0.0"/>
            <translate time = "10" align="True">
                <point x = "2.236068" y = "0" z = "0" />
                <point x = "1.58113" y = "0" z = "-1.58113" />
                <point x = "0" y = "0" z = "-2.236068" />
                <point x = "-1.58113" y = "0" z = "-1.58113" />
                <point x = "-2.236068" y = "0" z = "0" />
                <point x = "-1.58113" y = "0" z = "1.58113" />
                <point x = "0" y = "0" z = "2.236068" />
                <point x = "1.58113" y = "0" z = "1.58113" />
            </translate>
            <scale x="0.25" y="0.25" z="0.25"/>
        </transform>
        <models>
            <model file="sphere.3d"/>
        </models>
    </group>
</group>
<group>
```

```
<!--MARTE-->
<transform>
    <color r="0.85" g="0.46" b="0.007"/>
    <translate time = "20" align="True">
        <point x = "6" y = "0" z = "0" />
        <point x = "4.24264" y = "0" z = "-4.24264" />
        <point x = "0" y = "0" z = "-6" />
        <point x = "-4.24264" y = "0" z = "-4.24264" />
        <point x = "-6" y = "0" z = "0" />
        <point x = "-4.24264" y = "0" z = "4.24264" />
        <point x = "0" y = "0" z = "6" />
        <point x = "4.24264" y = "0" z = "4.24264" />
    </translate>
    <rotate time="3" x="0" y="1" z="0" />
    <rotate angle="260" x="0" y="1" z="0"/>
    <scale x="0.15" y="0.15" z="0.15"/>
</transform>
<models>
    <model file="sphere.3d"/>
</models>
<group>
    <!--LUA1-->
    <transform>
        <color r="1.0" g="1.0" b="1.0"/>
        <rotate angle="-38.66" x="1" y="0.0" z="0.0"/>
        <translate time = "10" align="True">
            <point x = "1.600781" y = "0" z = "0" />
            <point x = "1.13192" y = "0" z = "-1.13192" />
            <point x = "0" y = "0" z = "-1.600781" />
            <point x = "-1.13192" y = "0" z = "-1.13192" />
            <point x = "-1.600781" y = "0" z = "0" />
            <point x = "-1.13192" y = "0" z = "1.13192" />
            <point x = "0" y = "0" z = "1.600781" />
            <point x = "1.13192" y = "0" z = "1.13192" />
        </translate>
        <scale x="0.25" y="0.25" z="0.25"/>
    </transform>
    <models>
        <model file="sphere.3d"/>
    </models>
</group>
<group>
    <!--LUA2-->
    <transform>
        <color r="0.82" g="0.73" b="0.55"/>
        <rotate angle="33.69" x="1" y="0.0" z="0.0"/>
        <translate time = "10" align="True">
            <point x = "1.802776" y = "0" z = "0" />
            <point x = "1.27475" y = "0" z = "-1.27475" />
            <point x = "0" y = "0" z = "-1.802776" />
            <point x = "-1.27475" y = "0" z = "-1.27475" />
            <point x = "-1.802776" y = "0" z = "0" />
            <point x = "-1.1.27475" y = "0" z = "1.27475" />
            <point x = "0" y = "0" z = "1.802776" />
```

```
                <point x = "1.27475" y = "0" z = "1.27475" />
            </translate>
            <scale x="0.3" y="0.3" z="0.3"/>
        </transform>
        <models>
            <model file="sphere.3d"/>
        </models>
    </group>
</group>
<group>
    <!--JUPITER-->
    <transform>
        <color r="1.0" g="0.74" b="0.48"/>
        <translate time = "30" align="True">
            <point x = "8" y = "0" z = "0" />
            <point x = "5.65685" y = "0" z = "-5.65685" />
            <point x = "0" y = "0" z = "-8" />
            <point x = "-5.65685" y = "0" z = "-5.65685" />
            <point x = "-8" y = "0" z = "0" />
            <point x = "-5.65685" y = "0" z = "5.65685" />
            <point x = "0" y = "0" z = "8" />
            <point x = "5.65685" y = "0" z = "5.65685" />
        </translate>
        <rotate time="6" x="0" y="1" z="0" />
        <rotate angle="150" x="0" y="1" z="0"/>
        <scale x="0.7" y="0.7" z="0.7"/>
    </transform>
    <models>
        <model file="sphere.3d"/>
    </models>
    <group>
        <!--LUA1-->
        <transform>
            <color r="0.82" g="0.27" b="0.0"/>
            <rotate angle="0.1" x="1" y="0.0" z="0.0"/>
            <translate time = "10" align="True">
                <point x = "2" y = "0" z = "0" />
                <point x = "1.41421" y = "0" z = "-1.41421" />
                <point x = "0" y = "0" z = "-2" />
                <point x = "-1.41421" y = "0" z = "-1.41421" />
                <point x = "-2" y = "0" z = "0" />
                <point x = "-1.41421" y = "0" z = "1.41421" />
                <point x = "0" y = "0" z = "2" />
                <point x = "1.41421" y = "0" z = "1.41421" />
            </translate>
            <scale x="0.1" y="0.1" z="0.1"/>
        </transform>
        <models>
            <model file="sphere.3d"/>
        </models>
    </group>
    <group>
        <!--LUA2-->
        <transform>
```

```
            <color r="1.0" g="1.0" b="1.0"/>
            <rotate angle="-25" x="1" y="0.0" z="0.0"/>
            <translate time = "10" align="True">
                <point x = "1.732051" y = "0" z = "0" />
                <point x = "1.22474" y = "0" z = "-1.22474" />
                <point x = "0" y = "0" z = "-1.732051" />
                <point x = "-1.22474" y = "0" z = "-1.22474" />
                <point x = "-1.732051" y = "0" z = "0" />
                <point x = "-1.22474" y = "0" z = "1.22474" />
                <point x = "0" y = "0" z = "1.732051" />
                <point x = "1.22474" y = "0" z = "1.22474" />
            </translate>
            <scale x="0.1" y="0.1" z="0.1"/>
        </transform>
        <models>
            <model file="sphere.3d"/>
        </models>
    </group>
    <group>
        <!--LUA3-->
        <transform>
            <color r="0.92" g="0.8" b="0.38"/>
            <rotate angle="35.26" x="1" y="0.0" z="0.0"/>
            <translate time = "10" align="True">
                <point x = "1.732051" y = "0" z = "0" />
                <point x = "1.22474" y = "0" z = "-1.22474" />
                <point x = "0" y = "0" z = "-1.732051" />
                <point x = "-1.22474" y = "0" z = "-1.22474" />
                <point x = "-1.732051" y = "0" z = "0" />
                <point x = "-1.22474" y = "0" z = "1.22474" />
                <point x = "0" y = "0" z = "1.732051" />
                <point x = "1.22474" y = "0" z = "1.22474" />
            </translate>
            <scale x="0.15" y="0.15" z="0.15"/>
        </transform>
        <models>
            <model file="sphere.3d"/>
        </models>
    </group>
    <group>
        <!--LUA4-->
        <transform>
            <color r="0.55" g="0.42" b="0.26"/>
            <rotate angle="-35.26" x="1" y="0.0" z="0.0"/>
            <translate time = "10" align="True">
                <point x = "1.732051" y = "0" z = "0" />
                <point x = "1.22474" y = "0" z = "-1.22474" />
                <point x = "0" y = "0" z = "-1.732051" />
                <point x = "-1.22474" y = "0" z = "-1.22474" />
                <point x = "-1.732051" y = "0" z = "0" />
                <point x = "-1.22474" y = "0" z = "1.22474" />
                <point x = "0" y = "0" z = "1.732051" />
                <point x = "1.22474" y = "0" z = "1.22474" />
            </translate>
```

```
            <scale x="0.15" y="0.15" z="0.15"/>
        </transform>
        <models>
            <model file="sphere.3d"/>
        </models>
    </group>
</group>
<group>
    <!--SATURNO-->
    <transform>
        <color r="0.56" g="0.83" b="0.42"/>
        <translate time = "40" align="True">
            <point x = "9.5" y = "0" z = "0" />
            <point x = "6.71751" y = "0" z = "-6.71751" />
            <point x = "0" y = "0" z = "-9.5" />
            <point x = "-6.71751" y = "0" z = "-6.71751" />
            <point x = "-9.5" y = "0" z = "0" />
            <point x = "-6.71751" y = "0" z = "6.71751" />
            <point x = "0" y = "0" z = "9.5" />
            <point x = "6.71751" y = "0" z = "6.71751" />
        </translate>
        <rotate time="5" x="0" y="1" z="0" />
        <rotate angle="-50" x="0" y="1" z="0"/>
        <scale x="0.65" y="0.65" z="0.65"/>
    </transform>
    <models>
        <model file="sphere.3d"/>
    </models>
    <group>
        <!--ANEL-->
        <transform>
            <rotate angle="-30" x="1" y="0.0" z="0.0"/>
        </transform>
        <models>
            <model file="torus.3d"/>
        </models>
    </group>
    <group>
        <!--LUA1-->
        <transform>
            <color r="0.48" g="0.63" b="0.38"/>
            <rotate angle="45" x="1" y="0.0" z="0.0"/>
            <translate time = "10" align="True">
                <point x = "1.414214" y = "0" z = "0" />
                <point x = "1" y = "0" z = "-1" />
                <point x = "0" y = "0" z = "-1.414214" />
                <point x = "-1" y = "0" z = "-1" />
                <point x = "-1.414214" y = "0" z = "0" />
                <point x = "-1" y = "0" z = "1" />
                <point x = "0" y = "0" z = "1.414214" />
                <point x = "1" y = "0" z = "1" />
            </translate>
            <scale x="0.15" y="0.15" z="0.15"/>
        </transform>
```

```
            <models>
                <model file="sphere.3d"/>
            </models>
        </group>
        <group>
            <!--LUA2-->
            <transform>
                <color r="1.0" g="1.0" b="1.0"/>
                <rotate angle="33.69" x="1" y="0.0" z="0.0"/>
                <translate time = "10" align="True">
                    <point x = "1.802776" y = "0" z = "0" />
                    <point x = "1.27475" y = "0" z = "-1.27475" />
                    <point x = "0" y = "0" z = "-1.802776" />
                    <point x = "-1.27475" y = "0" z = "-1.27475" />
                    <point x = "-1.802776" y = "0" z = "0" />
                    <point x = "-1.27475" y = "0" z = "1.27475" />
                    <point x = "0" y = "0" z = "1.802776" />
                    <point x = "1.27475" y = "0" z = "1.27475" />
                </translate>
                <scale x="0.1" y="0.1" z="0.1"/>
            </transform>
            <models>
                <model file="sphere.3d"/>
            </models>
        </group>
        <group>
            <!--LUA3-->
            <transform>
                <color r="0.5" g="0.31" b="0.0"/>
                <rotate angle="-56.31" x="1" y="0.0" z="0.0"/>
                <translate time = "10" align="True">
                    <point x = "1.802776" y = "0" z = "0" />
                    <point x = "1.27475" y = "0" z = "-1.27475" />
                    <point x = "0" y = "0" z = "-1.802776" />
                    <point x = "-1.27475" y = "0" z = "-1.27475" />
                    <point x = "-1.802776" y = "0" z = "0" />
                    <point x = "-1.27475" y = "0" z = "1.27475" />
                    <point x = "0" y = "0" z = "1.802776" />
                    <point x = "1.27475" y = "0" z = "1.27475" />
                </translate>
                <scale x="0.15" y="0.15" z="0.15"/>
            </transform>
            <models>
                <model file="sphere.3d"/>
            </models>
        </group>
    </group>
    <group>
        <!--URANO-->
        <transform>
            <color r="0.35" g="0.75" b="0.86"/>
            <translate time = "50" align="True">
                <point x = "10.5" y = "0" z = "0" />
                <point x = "7.42462" y = "0" z = "-7.42462" />
```

```
            <point x = "0" y = "0" z = "-10.5" />
            <point x = "-7.42462" y = "0" z = "-7.42462" />
            <point x = "-10.5" y = "0" z = "0" />
            <point x = "-7.42462" y = "0" z = "7.42462" />
            <point x = "0" y = "0" z = "10.5" />
            <point x = "7.42462" y = "0" z = "7.42462" />
        </translate>
        <rotate time="2.5" x="0" y="-1" z="0"/>
        <rotate angle="39" x="0" y="1" z="0"/>
        <scale x="0.25" y="0.25" z="0.25"/>
    </transform>
    <models>
        <model file="sphere.3d"/>
    </models>
    <group>
        <!--LUA1-->
        <transform>
            <color r="1.0" g="1.0" b="1.0"/>
            <rotate angle="29.02" x="1" y="0.0" z="0.0"/>
            <translate time = "10" align="True">
                <point x = "2.061553" y = "0" z = "0" />
                <point x = "1.45773" y = "0" z = "-1.45773" />
                <point x = "0" y = "0" z = "-2.061553" />
                <point x = "-1.45773" y = "0" z = "-1.45773" />
                <point x = "-2.061553" y = "0" z = "0" />
                <point x = "-1.45773" y = "0" z = "1.45773" />
                <point x = "0" y = "0" z = "2.061553" />
                <point x = "1.45773" y = "0" z = "1.45773" />
            </translate>
            <scale x="0.15" y="0.15" z="0.15"/>
        </transform>
        <models>
            <model file="sphere.3d"/>
        </models>
    </group>
    <group>
        <!--LUA2-->
        <transform>
            <color r="0.87" g="0.59" b="0.28"/>
            <rotate angle="56.31" x="1" y="0.0" z="0.0"/>
            <translate time = "10" align="True">
                <point x = "1.802776" y = "0" z = "0" />
                <point x = "1.27475" y = "0" z = "-1.27475" />
                <point x = "0" y = "0" z = "-1.802776" />
                <point x = "-1.27475" y = "0" z = "-1.27475" />
                <point x = "-1.802776" y = "0" z = "0" />
                <point x = "-1.27475" y = "0" z = "1.27475" />
                <point x = "0" y = "0" z = "1.802776" />
                <point x = "1.27475" y = "0" z = "1.27475" />
            </translate>
            <scale x="0.25" y="0.25" z="0.25"/>
        </transform>
        <models>
            <model file="sphere.3d"/>
```

```
        </models>
    </group>
</group>
<group>
    <!--NEPTUNO-->
    <transform>
        <color r="0.11" g="0.33" b="0.58"/>
        <translate time = "100" align="True">
            <point x = "12" y = "0" z = "0" />
            <point x = "8.48528" y = "0" z = "-8.48528" />
            <point x = "0" y = "0" z = "-12" />
            <point x = "-8.48528" y = "0" z = "-8.48528" />
            <point x = "-12" y = "0" z = "0" />
            <point x = "-8.48528" y = "0" z = "8.48528" />
            <point x = "0" y = "0" z = "12" />
            <point x = "8.48528" y = "0" z = "8.48528" />
        </translate>
        <rotate time="2" x="0" y="1" z="0" />
        <rotate angle="15" x="0" y="1" z="0"/>
        <scale x="0.2" y="0.2" z="0.2"/>
    </transform>
    <models>
        <model file="sphere.3d"/>
    </models>
    <group>
        <!--LUA-->
        <transform>
            <color r="0.98" g="0.73" b="0.01"/>
            <rotate angle="-26.57" x="1" y="0.0" z="0.0"/>
            <translate time = "10" align="True">
                <point x = "2.236068" y = "0" z = "0" />
                <point x = "1.58113" y = "0" z = "-1.58113" />
                <point x = "0" y = "0" z = "-2.236068" />
                <point x = "-1.58113" y = "0" z = "-1.58113" />
                <point x = "-2.236068" y = "0" z = "0" />
                <point x = "-1.58113" y = "0" z = "1.58113" />
                <point x = "0" y = "0" z = "2.236068" />
                <point x = "1.58113" y = "0" z = "1.58113" />
            </translate>
            <scale x="0.25" y="0.25" z="0.25"/>
        </transform>
        <models>
            <model file="sphere.3d"/>
        </models>
    </group>
</group>
<group>
    <!--COMETA-->
    <transform>
        <color r="0.5" g="0.31" b="0.0"/>
        <translate time = "20" align="True">
            <point x = "-1" y = "-3" z = "-4" />
            <point x = "-3" y = "0" z = "4" />
            <point x = "4" y = "3" z = "2" />
```

```
                <point x = "3" y = "0" z = "0" />
            </translate>
            <scale x="0.1" y="0.1" z="0.1"/>
        </transform>
        <models>
            <model file="bezier.3d"/>
        </models>
    </group>
</world>
```