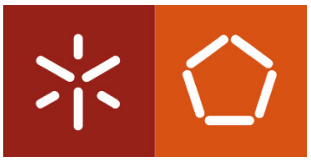
Universidade do Minho

Escola de Engenharia



Computação Gráfica

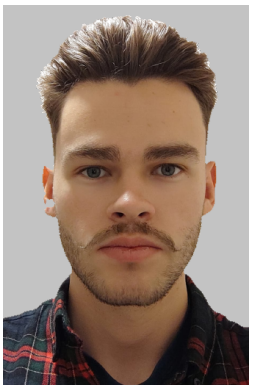
Licenciatura em Engenharia Informática

Relatório do TP2

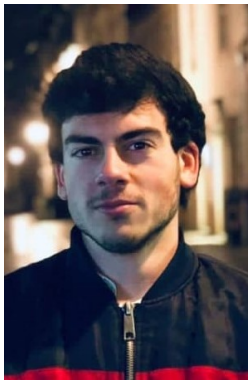
Geometric Transforms

Grupo 32

José Magalhães **A93273**



Carlos Dias **A93185**



Francisco Izquierdo **A93170**

Duarte Lucas **A93170**

Fevereiro, 2023

Conteúdo

Lista de Figuras

1. Introdução

1.1 Contextualização

Nesta segunda fase, foi proposto que fosse processado um ficheiro XML com uma hierarquia de transformações geométricas e modelos de forma que o motor gráfico renderize uma determinada cena.

1.2 Objetivos e Trabalho Proposto

O presente relatório tem em vista formalizar toda a análise e modelação pretendida face ao problema proposto, incluindo estruturas de dados.

Para esta segunda fase, foi proposto representar um sistema solar estático num ficheiro XML utilizando modelos de primitivas construídas na primeira fase (esfera) e um conjunto de transformações geométricas, de forma que o motor gráfico finalize uma determinada cena.

1.3 Resumo do Trabalho a Desenvolver

De maneira a poder finalizar cenas customizadas em concordância com um ficheiro XML, será necessário fazer uma atualização no motor gráfico desenvolvido na primeira fase, como a sua capacidade de processamento de ficheiros XML e a estrutura de dados para armazenar informações necessárias para finalizar as cenas.

2. Metodologia de Resolução

2.1 Atualização do Generator

Nesta segunda parte foi necessário fazer algumas alterações tanto no *generator* como no *engine*, tal como iremos abordar posteriormente, de modo a cumprir todos os requisitos para esta fase do projeto curricular.

2.1.1 Torus

As primitivas anteriormente feitas não iriam suportar na totalidade a construção de um sistema solar, tal como era pretendido. De tal modo, o grupo de trabalho teve de acrescentar a primitiva *Torus* na qual recebe como parâmetros raio interior, raio exterior, o número de *slices* e o número de *rings*.

A construção desta primitiva passou inicialmente por fazer o tratamento dos parâmetros recebidos, isto é, ao receber o número de *slices* e de *rings* conseguimos obter o ângulo de intervalo que deveremos utilizar para construir as circunferências dos anéis do Torus e posteriormente onde posicioná-los. Após este tratamento de dados, é feito dois *whiles* consecutivos, em que o primeiro é responsável por fazer a criação de um círculo de "referência"que será posteriormente utilizado no segundo *while*. Por sua vez, o segundo *while* é acompanhado por um *for*, um ciclo interior, em que a cada iteração deste são desenhados dois triângulos delimitados pelos parâmetros anteriores, isto é, os últimos ciclos rodam e posicionam a circunferência de referência de modo a formar o Torus.

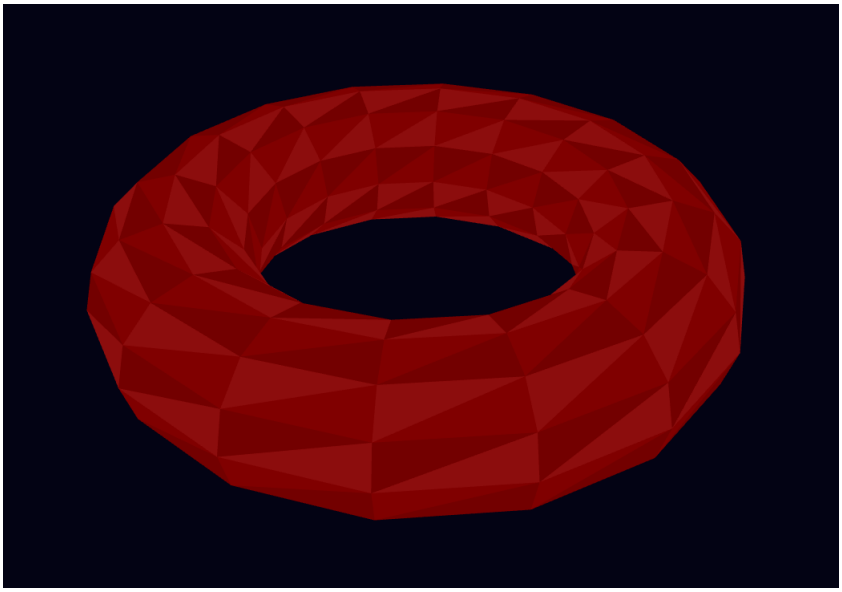


Figura 2.1: Torus

2.2 Atualização do Engine

Em relação ao *engine*, de forma a conseguir a ler os novos formatos do ficheiro XML propostos nesta segunda fase, foi necessário que este sofresse alterações no modo de leitura dos mesmos e consequentemente a estrutura de dados associada.

2.3 Estrutura de Dados

A estrutura de dados foi um dos parâmetros que sofreu uma grande alteração, devido à leitura dos novos ficheiros XML. Deste modo, passamos a ter uma lista de objetos, em que cada um deles é constituído por uma lista de pontos, uma posição, uma escala, a cor e um *boolean* que indica se este é centralizável ou não. Cada objeto contêm ainda uma lista de transformações, em que cada transformação tem um ponto e um tipo, podendo este ser uma translação, escalar, rotação. Por fim, continuamos com a mesma estrutura para os pontos, isto é, cada um deles é constituído por 3 coordenadas.

```
struct point{
    float x = 0;
    float y = 0;
    float z = 0;
};

struct transform{
    point p;
    int type = -1;
};

struct object{
    std::list<point> vertices;
    std::list<transform> transforms;
    point position;
    point scale;
    point color;
    bool centerable = false;
};
```

Figura 2.2: Structs

2.4 Processamento de Ficheiros XML

Para podermos realizar o processamento do ficheiro XML, foi necessário fazer uso da biblioteca TinyXML, que permite fazer ordenadamente a leitura em árvore do ficheiro. O nosso trabalho possui três ficheiros XML. Dois dos ficheiros representam o sistema solar como um todo, em contrapartida, o outro ficheiro apresenta todas as primitivas individualmente.

O ficheiro XML irá conter como componentes os seguintes elementos:

- Grupo

- Models
- Position
- Rotation
- Scale

2.4.1 Grupo

Certos elementos e atributos necessitaram de ser agrupados fazendo uso do group em XML. Alguns dos grupos foram representados por um nome único. Temos o grupo principal denominado por Sistema Solar, logo depois temos o grupo com o nome Sol, que dentro deste foram definidos outros grupos que representam os diversos planetas, luas e as orbitas de cada planeta.

Figura 2.3: Tag: Groups

2.4.2 Models

No ficheiro XML, a tag models aparece conectada aos ficheiros .3d. Esses ficheiros assinalam as figuras a serem renderizadas. É introduzido como parâmetro a cor pretendida para cada esfera no formato rgb.

De seguida é demonstrado um exemplo de como a tag models é usada no ficheiro XML:

```
<models>
    <model file="orbita_2.3d" r="0.8" g="0.8" b="0.8"/>
</models>
```

Figura 2.4: Tag: Models

2.4.3 Translation, Rotation, Scale

As tags translate, rotate e scale inserem-se na tag transform, que estão associadas à translação, rotação e escala de um determinado planeta do nosso sistema solar com os respetivos valores em relação à posição atual, como mostra a figura seguinte:

```
<group name="planetas">
    <group name="orbitaMercurio">
        <transform>
            <rotation angle="1" x="1" y="0" z="0" />
            <rotation angle="45" x="0" y="1" z="0" />
        </transform>
        <group>
            <transform>
                <scale x="0.001725" y="0.001725" z="0.001725" />
            </transform>
            <models>
                <model file="orbita_2.3d" r="0.8" g="0.8" b="0.8"/>
            </models>
        </group>
        <group name="mercurio">
            <transform>
                <position x="1.2" y="0" z="1.2" />
                <scale x="0.07" y="0.07" z="0.07" />
            </transform>
            <models>
                <model file="sphere.3d" r="0.5" g="0.5" b="0.5" centerable="true"/>
            </models>
        </group>
    </group>
```

Figura 2.5: Tag: Translation, Rotation, Scale

3. Sistema Solar

De modo a demonstrar visualmente o programa desenvolvido pelo grupo de trabalho nesta segunda fase, foi nos pedido uma representação do sistema solar, que para a qual foi necessário a criação do ficheiro XML supramencionado e vários outros ficheiros .3d.

A equipa de trabalho procurou uma solução pouco realista, de modo a permitir uma fácil visualização de todos planetas. No entanto procuramos fazer com que os astros solares representados tivessem uma altura proporcional entre eles, na qual foi respeitada a ordem pela qual os planetas estão dispostos, procurando também manter uma distância proporcional entres os astros.

Com recurso ao gerador implementado na fase anterior, foram criados 5 tipos de ficheiros .3d com o propósito de servirem como modelos para o ficheiro XML. Deste modo, foi gerado o ficheiro planet.3d que por sua vez irá representar o Sol e todos os planetas e luas. Foi criado o ficheiro asteroid.3d para representar os asteróides, os ficheiros orbita_1.3d e orbita_2.3d para representar as orbitas dos planetas e luas e por fim, foi criado o ficheiro torus.3d dando origem aos anéis de Saturno.

De seguida são apresentadas figuras obtidas no fim da construção do nosso sistema solar.

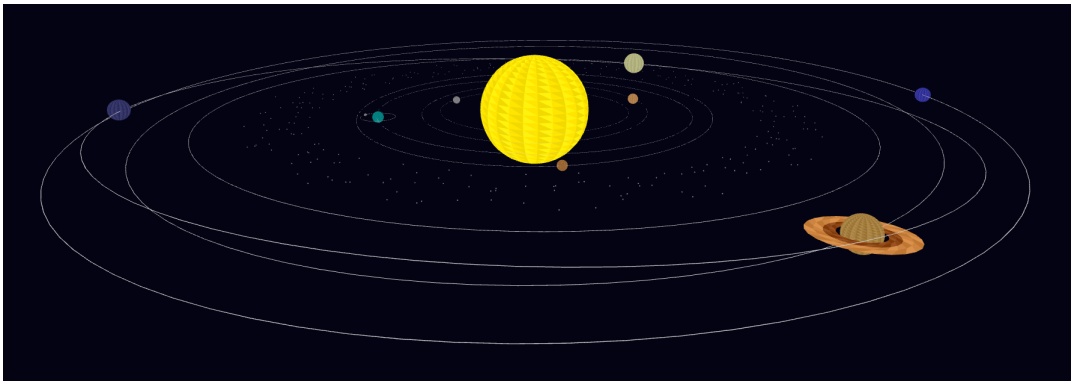


Figura 3.1: Sistema Solar

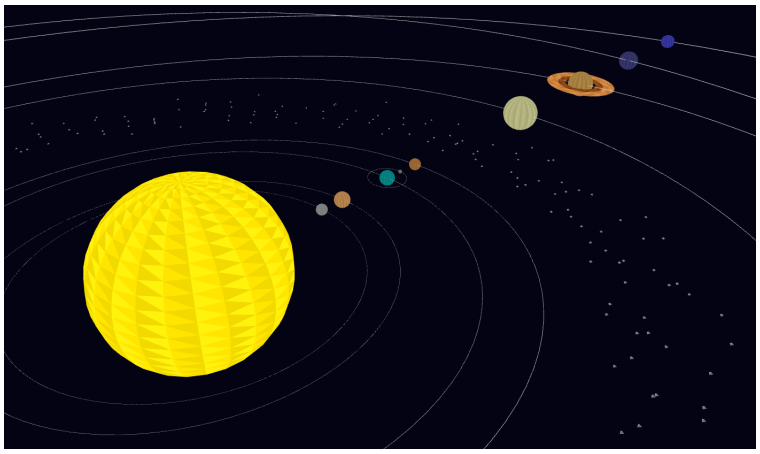


Figura 3.2: Sistema Solar - Alinhado

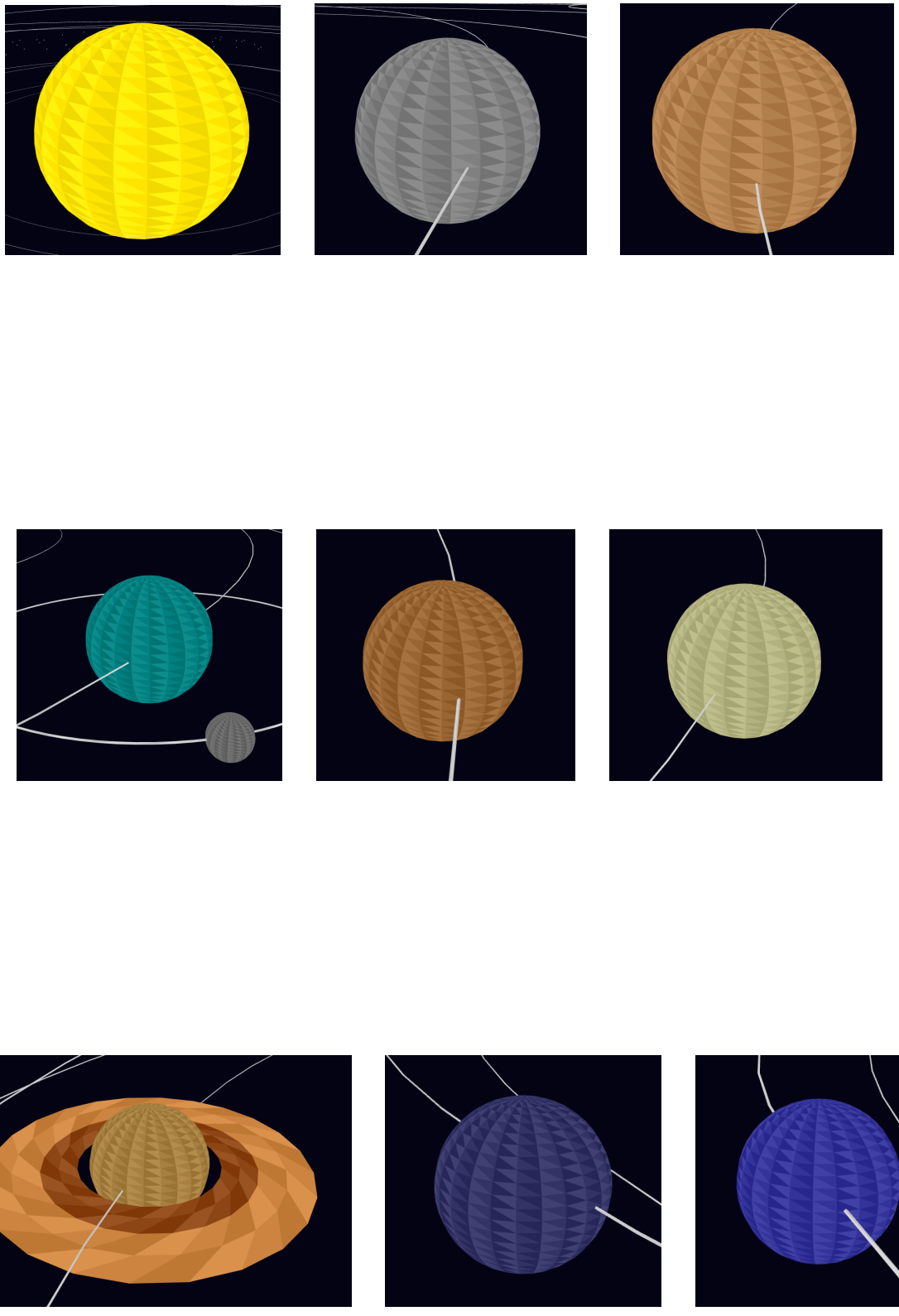


Figura 3.3: Planetas

4. Conclusão

Com esta segunda fase do projeto foi possível consolidar os vários conceitos abordados nas aulas práticas e teóricas, é importante salientar a vasta utilização do *OpenGL* que permitiu, por sua vez, um maior aprofundamento do nosso conhecimento.

Enquanto equipa de trabalho consideramos que esta fase foi concluída com sucesso, na medida em que é possível visualizar um sistema solar gerado com pormenores bastante aproximado das expectativas para esta segunda fase.