



**Universidade do Minho**

**Transformações Geométricas**  
**Unidade Curricular de Computação Gráfica**  
Licenciatura em Ciências da Computação  
Universidade do Minho

Bruno Jardim  
(A91680)

Inês Presa  
(A90355)

Tiago Carriço  
(A91695)

Tiago Leite  
(A91693)

31 de março de 2022

# Índice

<b>1</b>	<b>Contextualização</b>	<b>2</b>
1.1	Enunciado . . . . .	2
<b>2</b>	<b>Apresentação das soluções</b>	<b>3</b>
2.1	Cenas hierárquicas . . . . .	3
2.1.1	Demonstração do funcionamento do engine . . . . .	4
2.2	Ficheiro de configuração . . . . .	5
<b>3</b>	<b>Extras</b>	<b>9</b>
3.1	Implementação de câmara FPS . . . . .	9
<b>4</b>	<b>Conclusão</b>	<b>10</b>

# Capítulo 1

## Contextualização

No âmbito da unidade curricular de Computação Gráfica da Licenciatura em Ciências da Computação foi proposta o desenvolvimento em *OpenGL* de um motor gráfico genérico que terá como função a criação de um sistema solar. Desenvolvimento esse que deve ser composto por quatro etapas.

### 1.1 Enunciado

Nesta segunda etapa foi proposto:

- **Cenas hierárquicas**

Alteração do *engine* de modo a ler ficheiros XML com vários grupos dispostos hierarquicamente. Um grupo pode ser composto por várias transformações geométricas (translação, rotação e escala), onde a ordem é relevante, e opcionalmente por um ou mais modelos.

- **Ficheiro de configuração**

Criação de um ficheiro de configuração XML que recrie um modelo estático do sistema solar, composto pelo sol, planetas, e luas dispostas hierarquicamente.

## Capítulo 2

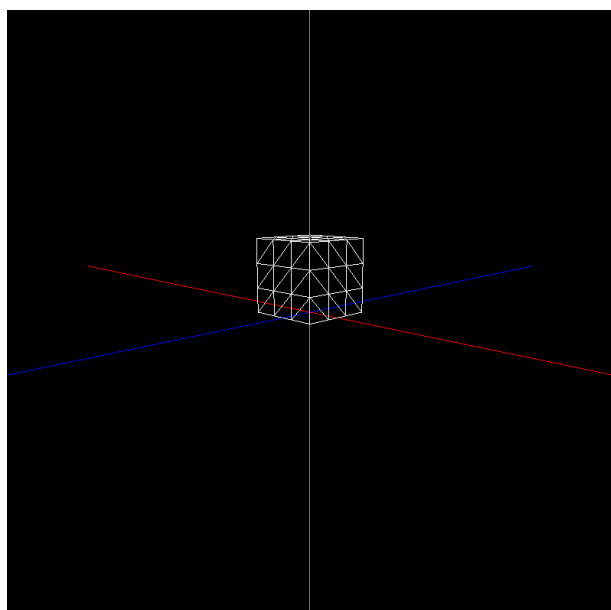
# Apresentação das soluções

### 2.1 Cenas hierárquicas

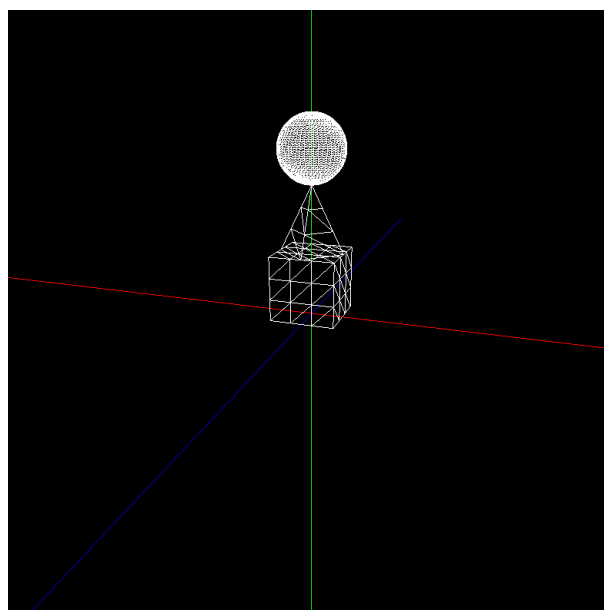
Para a criação de cenas hierárquicas, procedeu-se à alteração do leitor XML desenvolvido na primeira fase. Assim sendo, em oposição ao método iterativo usado anteriormente, nesta nova versão, o leitor trabalha de forma recursiva, análogamente a uma travessia em profundidade. Para tal, foi construída uma função auxiliar que recebe como argumentos o grupo a ler e um vetor com as transformações que vai herdar. Após a leitura de cada modelo, este é armazenado num vetor de modelos, que é uma estrutura composta por um vetor de pontos para desenhar a primitiva gráfica, e um vetor de transformações a aplicar antes da primitiva ser desenhada. De seguida, a função é chamada novamente recebendo como parâmetros o grupo descendente do atual e um vetor de transformações atualizado. Quando a função retorna do processo recursivo para esta etapa, o processo é repetido para o grupo "irmão" do atual.

Para a execução das transformações foi criada uma classe abstracta chamada *Transformation* com um método *apply* e mais três outras classes, *Translation*, *Rotation*, e *Scale*, que implementam esta classe. Desta forma, para cada primitiva é possível iterar pelo vetor que armazena as transformações e simplesmente invocar o método *apply*, não sendo por isso necessário executar instruções condicionais, como por exemplo *if else*.

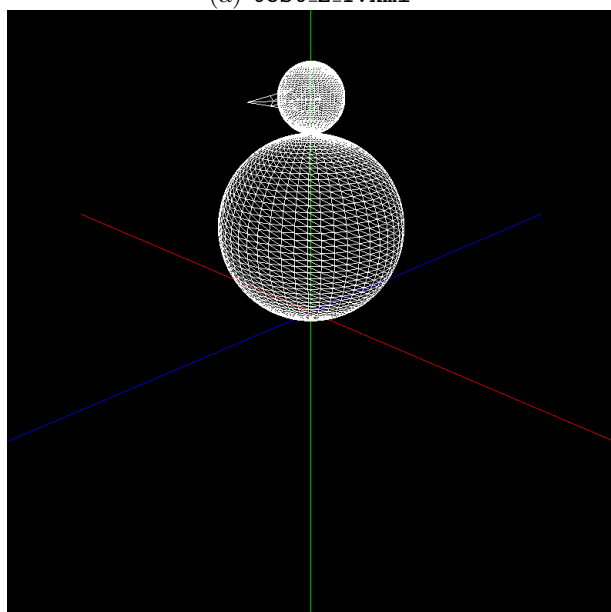
### 2.1.1 Demonstração do funcionamento do engine



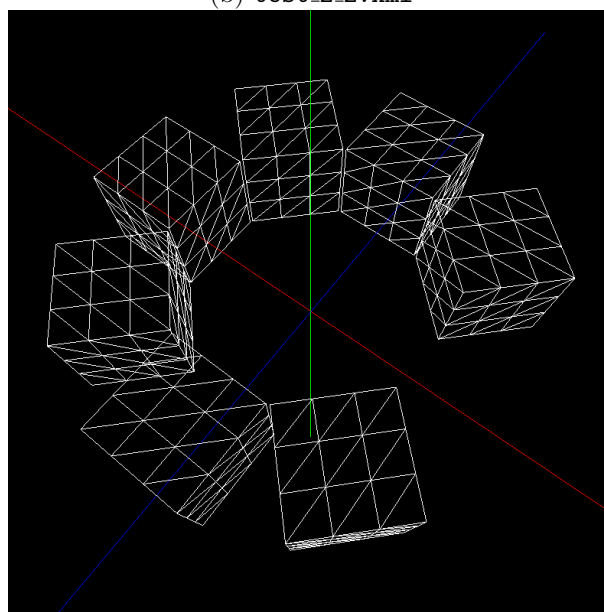
(a) test\_2.1.xml



(b) test\_2.2.xml



(c) test\_2.3.xml



(d) test\_2.4.xml

Figura 2.1: Execução dos ficheiros XML de teste

**Nota:** Nas imagens de teste apresentadas as esferas foram criadas com um número de *stacks* e *slices* superiores ao sugerido no ficheiro de configuração.

## 2.2 Ficheiro de configuração

No ficheiro de configuração XML colocou-se o sol, os planetas e algumas das luas e anéis que orbitam os planetas.

Uma vez que usando uma escala que representasse fielmente as distâncias e tamanhos tornaria inviável a visualização de todos os componentes do Sistema, decidiu-se implementar da seguinte forma:

- **Tamanho dos objetos:** usou-se como medida de escala o diâmetro da Terra.
- **Distância entre objetos:** utilizou-se a seguinte escala  $10^6 \text{ km} \rightarrow 1$  unidade.

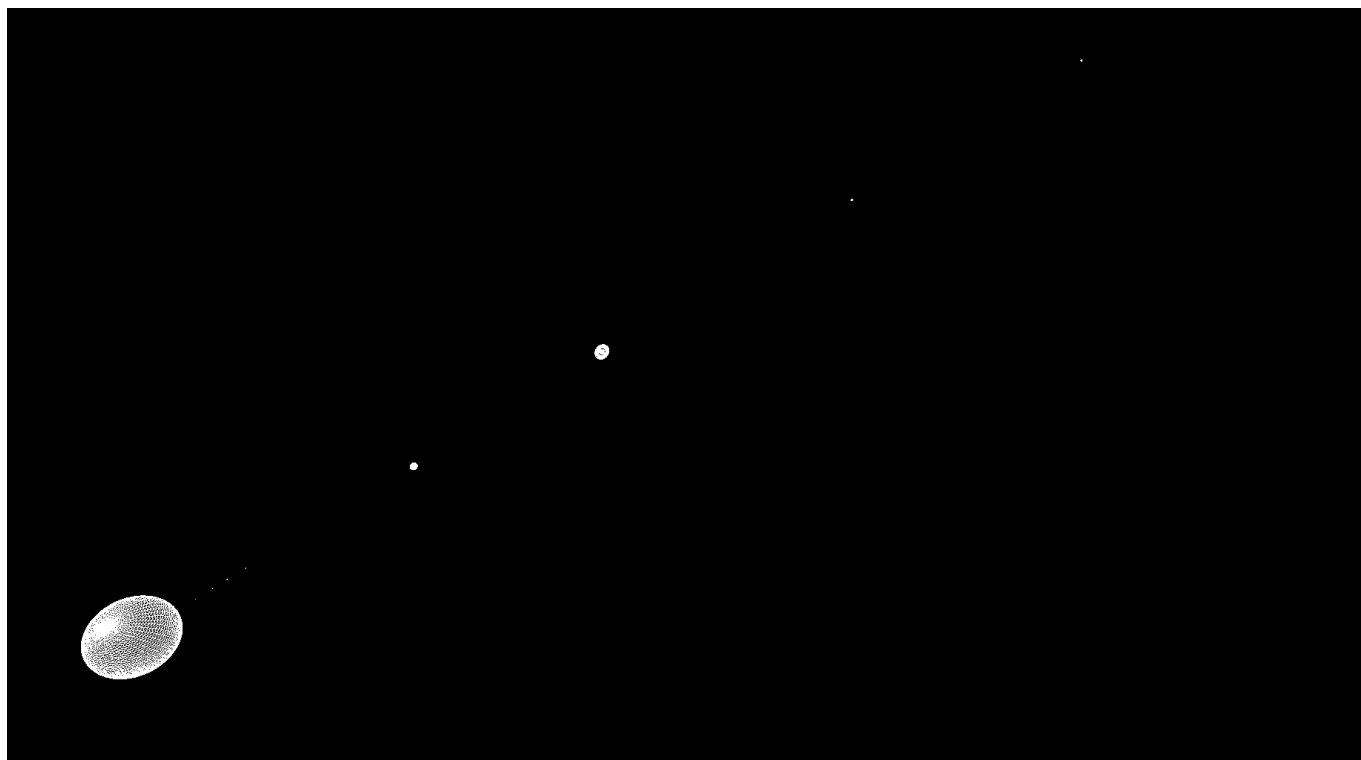


Figura 2.2: Vista global

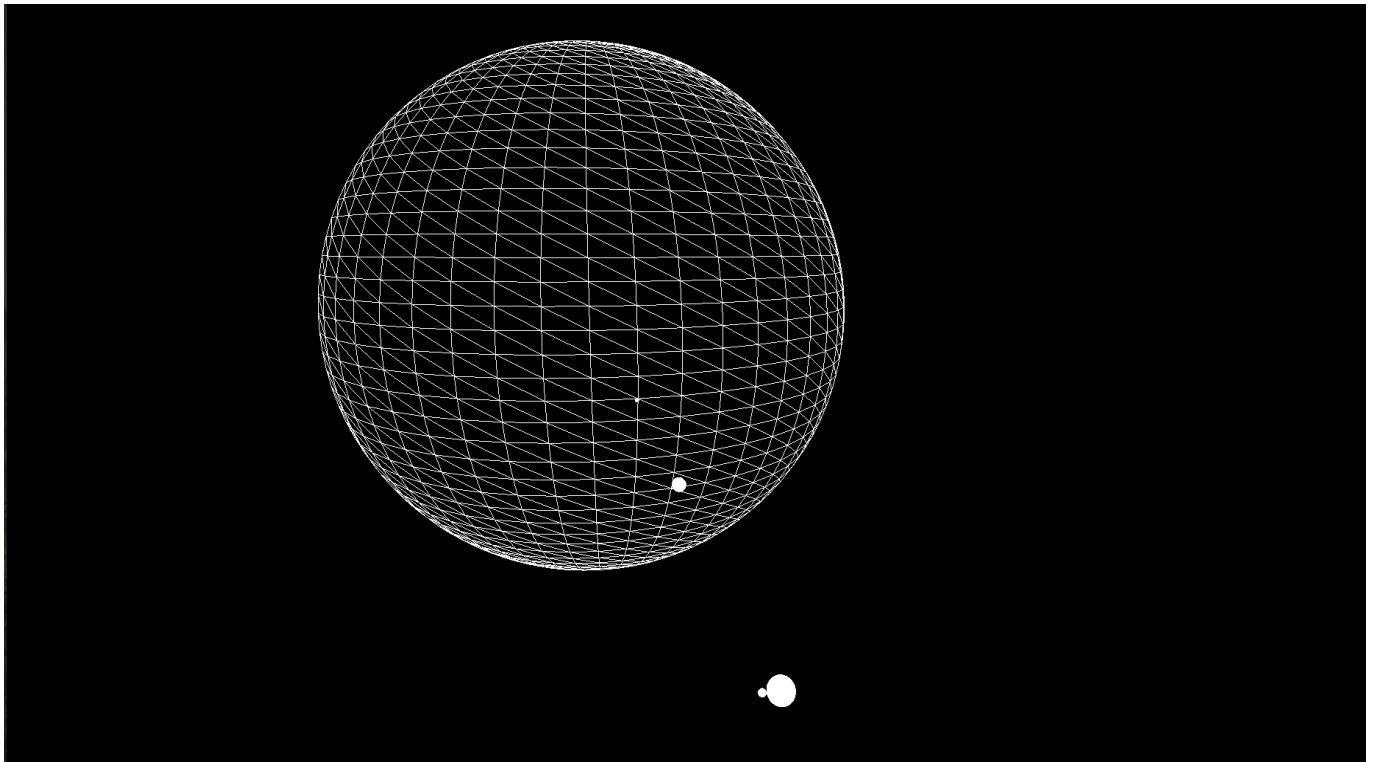


Figura 2.3: Sol - Terra

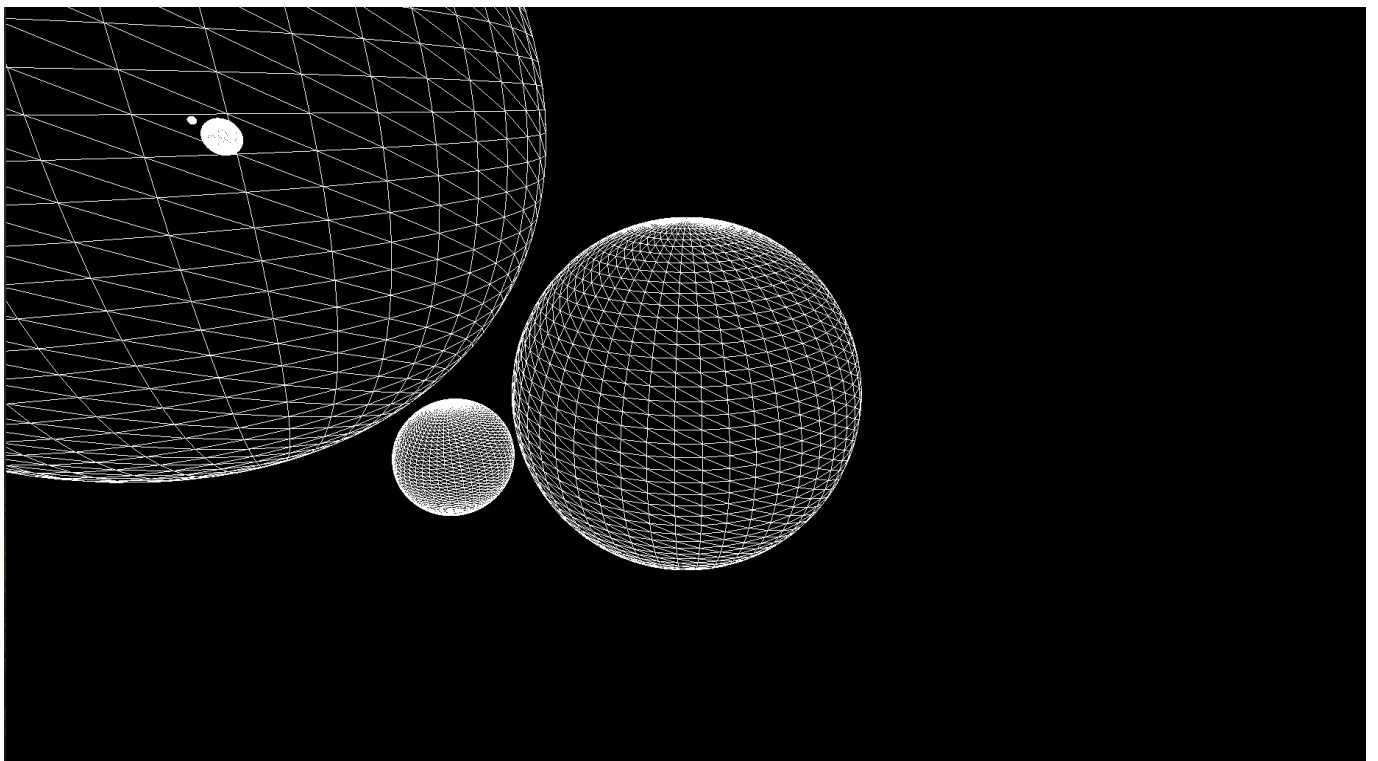


Figura 2.4: Terra - Lua

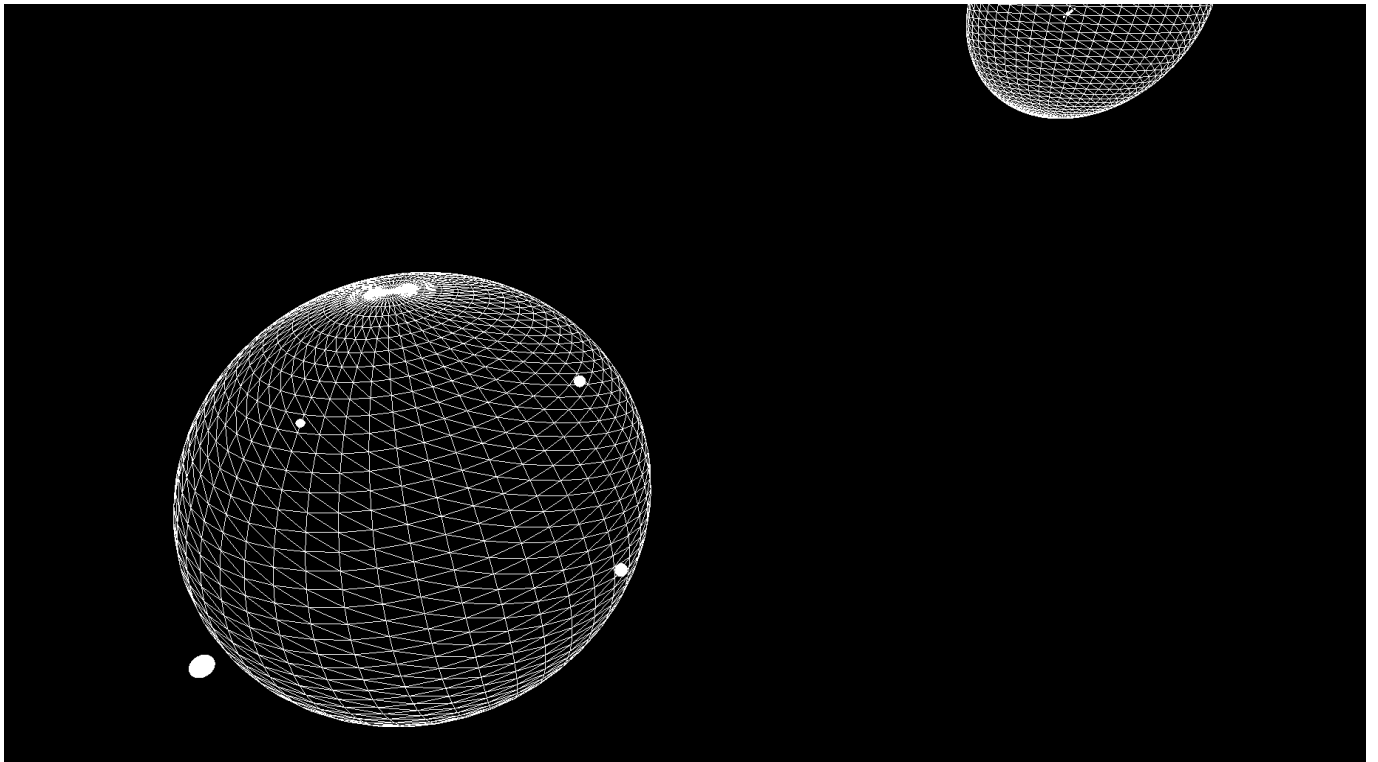


Figura 2.5: Júpiter com as luas de Galileu

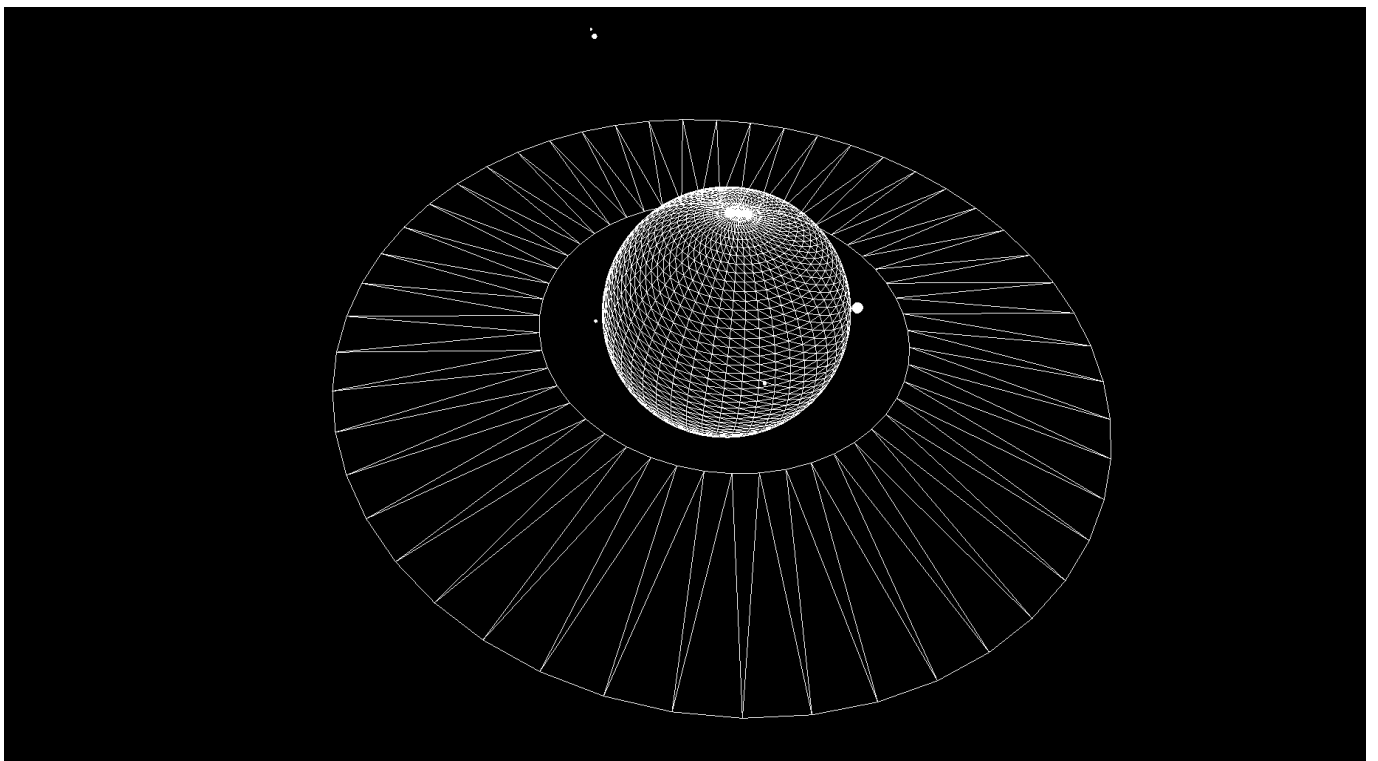


Figura 2.6: Saturno e as luas : Titan, Rhea e Iapetus



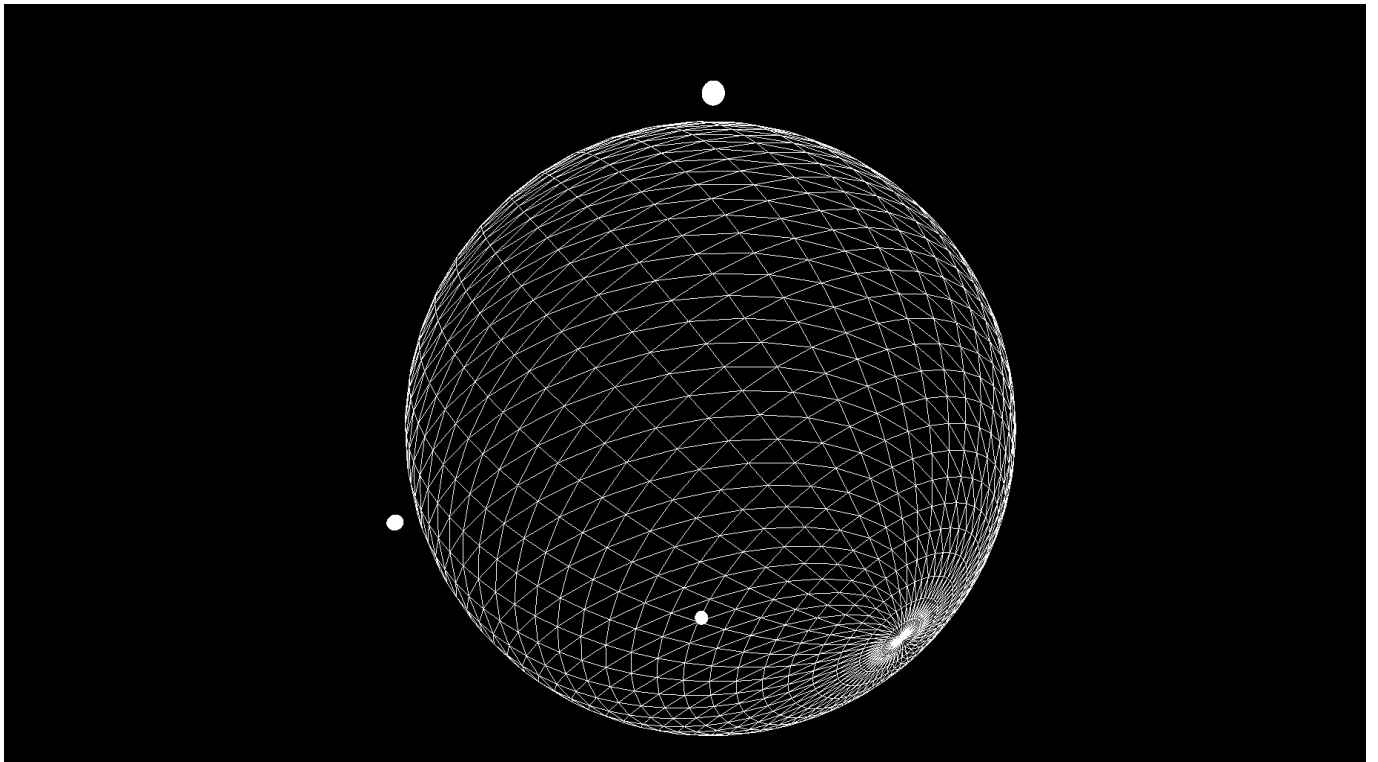


Figura 2.7: Úrano com as suas luas: Titania, Oberon e Umbriel

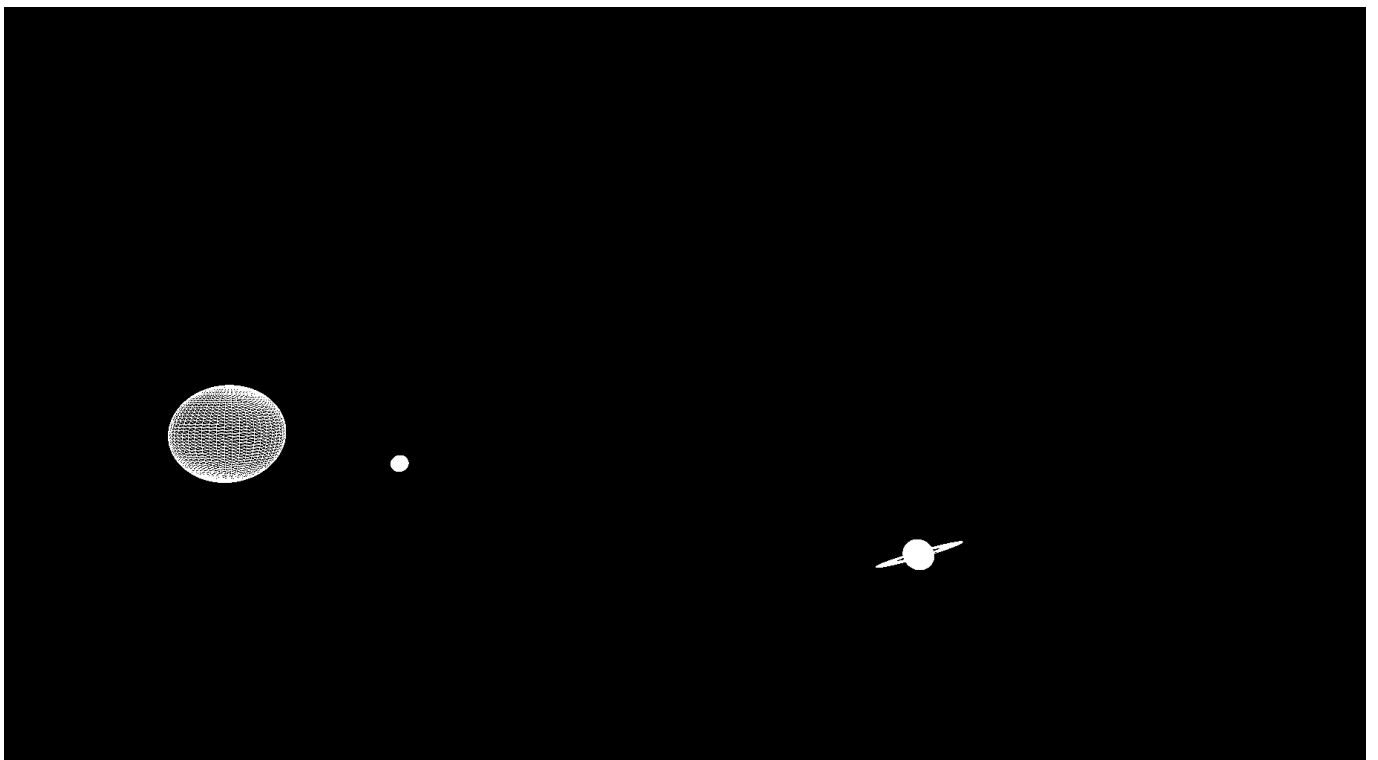


Figura 2.8: Sol - Saturno

## Capítulo 3

# Extras

### 3.1 Implementação de câmara FPS

Para ser possível viajar pelo Universo e dessa forma ”visitar” os diferentes planetas, tomou-se a decisão de implementar uma *câmara FPS*. Esta permite rodar a câmara com as *arrowkeys* e movimentar, para a frente, trás, esquerda e direita, com as teclas **w**, **s**, **a**, **d**, respetivamente.

Para demonstrar esta função encontra-se um vídeo em anexo.

## Capítulo 4

# Conclusão

A principal dificuldade enfrentada durante esta fase do trabalho foi a alteração da função que faz a leitura de ficheiros XML. A forma como idealizamos o seu funcionamento não foi muito difícil, no entanto durante a sua implementação depara-mo-nos com alguns problemas de *segmentation fault*, assim como transformações que ocorriam em locais inesperados.

Um fator que também criou dificuldades foi a escolha de uma escala adequada para definir o tamanho das diferentes componentes do Sistema Solar e a distância entre elas. Apesar de assim ser mais realista, não temos a certeza se foi a melhor opção.

Por último, a alteração da câmara por forma a permitir "viajar" pelo sistema solar também não foi tarefa fácil e pensamos que a mesma ainda pode ser melhorada.

Contudo, apesar das dificuldades enfrentadas, podemos concluir que os objectivos foram atingidos.