



Universidade do Minho
Departamento de Informática

Computação Gráfica

Ano Letivo 2023/2024

Trabalho Prático Fase 2 - Transformações Geométricas

Grupo 13

Ema Martins - A97678

Gonçalo Costa - A100824

Henrique Malheiro - A97455

Marta Rodrigues - A100743

Índice

Introdução	3
Engine	3
Parsing de ficheiros XML	3
Desenho	4
Extra (Câmara Orbital)	4
Sistema Solar	5
Torus (Anel de Saturno)	5
Formato do ficheiro XML	7
Resultado para Sistema Solar	9
Extra (Casas)	10
Conclusão	12

Introdução

Este relatório foi elaborado no âmbito da segunda fase do projeto da unidade curricular de Computação Gráfica, que consistiu em estender o trabalho realizado na primeira fase, atualizando essencialmente o Engine. O foco destas alterações encontra-se na criação de cenas de forma hierárquica utilizando transformações geométricas (translações, rotações e escalas) e modelos, através de instruções introduzidas por ficheiros XML num determinado formato.

Como validação do trabalho desenvolvido, foi escrito um ficheiro XML capaz de desenhar um modelo estático do Sistema Solar, definido numa hierarquia. Adicionalmente, foram também implementados alguns extras, como uma câmara orbital dentro do Engine, novas primitivas geométricas no Generator.

As diferentes metodologias e abordagens implementadas encontram-se detalhas mais à frente neste relatório.

Engine

Resultante da alteração do formato dos ficheiros XML, no módulo Engine procedeu-se a uma alteração da lógica de parsing e do seu processamento em novas estruturas de dados.

Parsing de ficheiros XML

Com a introdução de tags <group>, foi necessário introduzir um parsing que percorre todas essas tags, bem como as suas tags <group> “filhas”.

Inicialmente, recorreu-se à criação de uma lista que guardaria todos os grupos na ordem que surgiam no ficheiro, independente ordem de relacionamento entre si, tendo a mesma todos os modelos, transformações e potenciais operações como criação e fecho de matrizes.

Posteriormente, prevendo o aumento de complexidade dos ficheiros XML em futuras fases do projeto, decidiu-se modificar o processamento da árvore de parsing, replicando o relacionamento entre as diversas tags, criando-se assim uma struct Group principal. Essa struct possui uma lista dos seus modelos, uma lista das transformações aplicadas e por fim uma lista dos seus grupos “filhos”, sendo esta lógica recursiva até aos grupos que não possuem descendência.

```
struct Group {  
    std::list<std::string> files;  
    std::list<Transformation> transformations;  
    std::list<Group> children;  
};
```

Figure 1: Estrutura Group

Nas transformações, optou-se também por recorrer ao auxílio de uma struct Transformation, que possui o tipo de transformação, que poderá ter valores definidos num enum Type para translação, rotação e escala. Além disso, a struct contém os valores dessa mesma transformação (em x, y, z e ângulo, caso existam).

```
enum Type{
    TRANSLATE,
    ROTATE,
    SCALE
};
struct Transformation {
    float x, y, z, angle = 0;
    Type type;
};
```

Figure 2: Estrutura Transformation

Desenho

Quando ocorre o desenho dos diversos modelos representados nos VBOs, aplica-se de maneira recursiva as transformações nos diversos grupos e seus modelos, sendo essas transformações aplicadas assim também nos seus grupos filhos. É importante salientar que no começo de cada novo group, realiza-se um `pushMatrix()`, sendo o `popMatrix()` apenas realizado no fecho da tag correspondente, garantindo então que só os seus filhos sejam afetados pelas suas transformações, e não outras possíveis tags group “irmãs” ou de qualquer outro tipo de relacionamento.





Extra (Câmera Orbital)

Para permitir manipular da câmera, foram criadas variáveis globais sendo essas *alfa*, *beta* e *raio*.

- $\text{alfa} = \arccos\left(\frac{z}{\sqrt{x^2 + z^2}}\right)$
- $\text{raio} = \sqrt{x^2 + y^2 + z^2}$
- $\text{beta} = \arcsin\left(\frac{y}{\sqrt{x^2 + y^2 + z^2}}\right)$

Sendo x,y e z as coordenadas da posição da câmera. A existência destas variáveis permite a que qualquer instante se obtenha a posição da câmera (*poscam*) atualizada, desde o momento da geração do desenho pelo *engine* até que sofra qualquer manipulação. Adicionalmente, foi implementada a capacidade de alterar a forma como os polígonos são renderizados. Estas manipulações possibilitam, assim, que se altere a vista dos modelos e seja possível tornar o sistema mais interativo.

Os comandos de manipulação da mesma via teclado são:

- **F1** - Zoom in
- **F2** - Zoom out
-  - Rotação para cima
-  - Rotação para baixo
-  - Rotação para a direita
-  - Rotação para a esquerda
- **F** - Muda o modo para GL_FILL
- **L** - Muda o modo para GL_LINE
- **P** - Muda o modo para GL_POINT

Sistema Solar

Para demonstrar as funcionalidades desenvolvidas nesta segunda fase foi exigido uma demonstração de um modelo estático do Sistema Solar. Assim, criou-se um ficheiro XML que define todas as transformações e primitivas necessárias, de forma hierárquica, para representar grande parte dos astros do Sistema Solar. Este ficheiro contém o Sol, todos os astros considerados planetas juntamente com alguns satélites naturais, nomeadamente a Lua e os satélites de Marte - Deimos e Fobos - e no caso de Saturno está também representado o seu anel característico.

Torus (Anel de Saturno)

De modo a complementar o modelo de Saturno decidimos adicionar-lhe o anel e, para tal, acrescentamos a primitiva *torus* ao generator. Para além de possuir a forma de um anel, esta primitiva pode ser “achatada” para atingir a forma que pretendíamos.

Para calcular os pontos de um torus deverão ser fornecidos o raio interior, o raio exterior e os números de slices e stacks pretendidas para o anel. Através dos raios fornecidos podemos determinar as medidas necessárias para a construção de um torus, ou seja, o $\text{raio do torus} = (\text{raio exterior} - \text{raio interior}) / 2$, que representa a média das duas distâncias, e a $\text{distância} = \text{raio do torus} - \text{raio interior}$, que fornece um valor correspondente a metade da largura do torus, tal como o esboço a seguir pretende exemplificar.

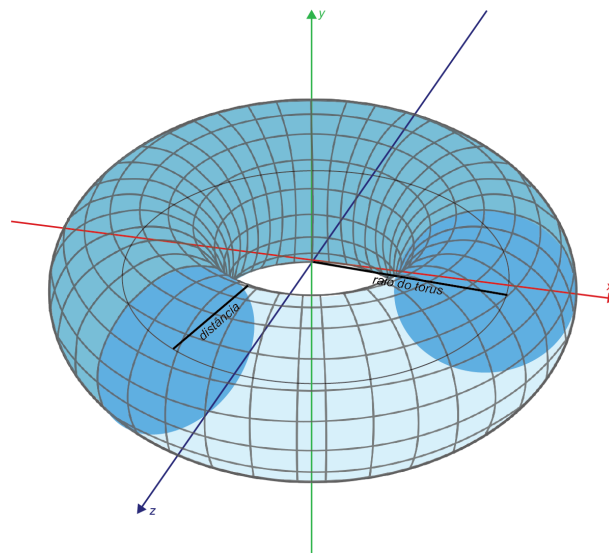


Figure 3: Esboço da geometria de um torus

Com estas informações, as coordenadas de um ponto no torus são obtidas da seguinte forma:

- $x = \text{distância} + \text{raio do torus} \times \cos(\beta) \times \cos(\alpha)$
- $y = \text{raio do torus} \times \sin(\beta)$
- $z = \text{distância} + \text{raio do torus} \times \cos(\beta) \times \sin(\alpha)$

Semelhante à primitiva da esfera, α um ângulo resultante da deslocação de slice a slice segundo um ângulo de tamanho $2\pi / \text{número de slices}$ e β obtido do deslocamento entre as stacks de $2\pi / \text{número de stacks}$. O torus é construído slice a slice aumentando o α e dentro de cada uma destas são desenhadas todas as suas stacks incrementando o β , gerando em cada passo 2 triângulos para os quais podemos inferir as seguintes coordenadas para os quatro pontos necessários.

- $p1 = ((\text{raio do torus} + \text{distância} \times \cos(\text{beta atual})) \times \cos(\text{alfa atual}), \text{distância} \times \sin(\text{beta atual}), (\text{raio do torus} + \text{distância} \times \cos(\text{beta atual})) \times \sin(\text{alfa atual}));$
- $p2 = ((\text{raio do torus} + \text{distância} \times \cos(\text{próximo beta})) \times \cos(\text{alfa atual}), \text{distância} \times \sin(\text{próximo beta}), (\text{raio do torus} + \text{distância} \times \cos(\text{próximo beta})) \times \sin(\text{alfa atual}));$
- $p3 = ((\text{raio do torus} + \text{distância} \times \cos(\text{próximo beta})) \times \cos(\text{próximo alfa}), \text{distância} \times \sin(\text{próximo beta}), (\text{raio do torus} + \text{distância} \times \cos(\text{próximo beta})) \times \sin(\text{próximo alfa}));$
- $p4 = ((\text{raio do torus} + \text{distância} \times \cos(\text{beta atual})) \times \cos(\text{próximo alfa}), \text{distância} \times \sin(\text{beta atual}), (\text{raio do torus} + \text{distância} \times \cos(\text{beta atual})) \times \sin(\text{próximo alfa}));$

Os triângulos necessários são inseridos pelas seguintes ordens: $[p1, p2, p3]$ e $[p1, p3, p4]$. Dando, assim, origem para cada paço a um quadrado semelhante ao seguinte:

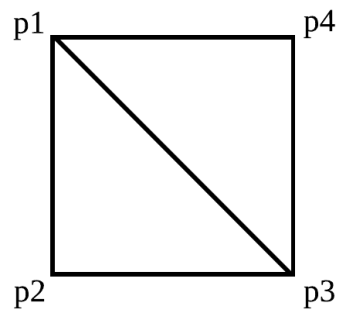


Figure 4: Formação de um quadrado pelos pontos do torus

A figura a seguir representa um exemplo de um torus desenhado a partir de um ficheiro .3d gerado para um torus:

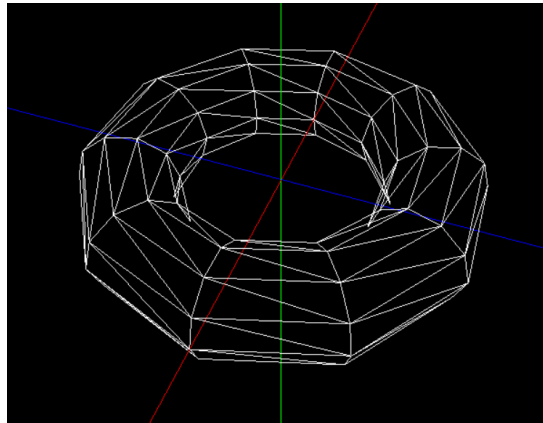


Figure 5: Exemplo de um Torus de raio interior 1, raio exterior 2, 10 slices e 10 stacks

Como o anel de Saturno possui um aspeto “espalmado”, determinamos que possui apenas 2 únicas stacks, uma possível ver por cima e outra de baixo, a figura é um exemplo com um aspeto semelhante à figura seguinte visto de cima de um torus com 8 slices e 2 stacks.

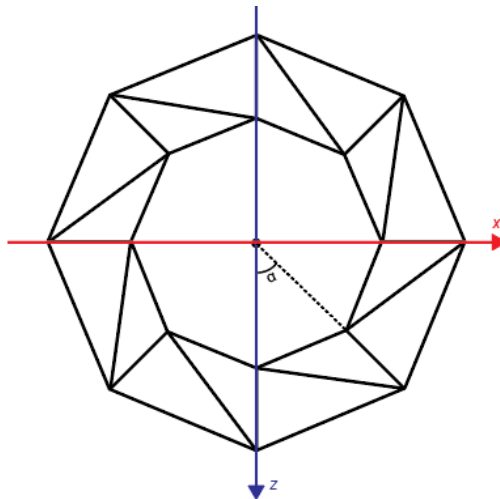


Figure 6: Desenho exemplo vista superior do anel de Saturno

Formato do ficheiro XML

O ficheiro XML para o desenho do Sistema Solar possui vários grupos que representam o Sol ou planetas, que podem conter subgrupos para representar certos astros, como a Lua relativa à Terra, Deimos e Fobos relativos a Marte e o anel de Saturno relativo a Saturno.

Para o desenho destes astros, foram gerados a partir do generator uma primitiva sphere e uma primitiva torus para os seus respetivos ficheiros .3d.

No caso da esfera foram utilizados os argumentos: raio = 1, stacks = 20, slices = 20, gerado no ficheiro sphere.3d; Já para o anel, foram utilizados os parâmetros: raio interior = 6, raio exterior = 8, slices = 20, stacks = 2, direcionado para o ficheiro torus.3d. Sendo este último utilizado para representar o anel de Saturno, as outras figuras foram desenhadas pela mesma esfera. Todos os astros foram aproximadamente escalados numa proporção do seu tamanho real para que a escala da Terra fosse 0,5.

O **Sol** foi desenhado no centro do referencial, sofrendo unicamente uma escala para o tornar maior.

```
<!-- SOL -->
<group>
  <transform>
    <scale x="20" y="20" z="20"/> <!-- Tamanho do Sol -->
  </transform>
  <models>
    <model file="sphere.3d" />
  </models>
</group>
```

Figure 7: Excerto do ficheiro XML para o desenho do Sol

Consoante o tamanho dos **planetas**, foram aplicadas escalas para tentar tornar o modelo o mais realista possível. Para além dessas transformações, foram aplicadas rotações relativas ao eixo Y para não se apresentarem todos alinhados e são aplicadas translações aos modelos para se distanciar os planetas relativamente ao Sol.

```
<!-- VÊNUS -->
<group>
  <transform>
    <rotate angle="89" x="0" y="1" z="0"/> <!-- Rotação relativa ao Sol de Vénus-->
    <translate x="29" y="0" z="0"/>
    <scale x="0.5" y="0.5" z="0.5"/> <!-- Tamanho de Vénus -->
  </transform>
  <models>
    <model file="sphere.3d" />
  </models>
</group>
```

Figure 8: Excerto do ficheiro XML para o desenho de Vénus

Relativamente aos **satélites naturais** apenas são desenhados os mesmos para a Terra e Marte. No caso da Lua e da Terra, por exemplo, é necessária uma translação para se posicionar relativamente ao Sol, e de seguida são usados dois subgrupos, um para desenhar a Terra e outro para desenhar a Lua. A Lua sofre uma pequena translação relativamente à posição da Terra. Estes mesmos passos aplicação similarmente a Marte, Deimos e Fobos.

```
<group>
  <transform>
    <rotate angle="185" x="0" y="1" z="0"/> <!-- Rotação relativa ao Sol da Terra-->
    <translate x="34" y="0" z="0"/>
  </transform>
  <!-- TERRA -->
  <group>
    <transform>
      <scale x="0.5" y="0.5" z="0.5"/> <!-- Tamanho da Terra -->
    </transform>
    <models>
      <model file="sphere.3d" />
    </models>
  </group>
  <!-- LUA -->
  <group>
    <transform>
      <translate x="0" y="0" z="1"/> <!-- Distância da Lua à Terra -->
      <scale x="0.1" y="0.1" z="0.1"/> <!-- Tamanho da Lua -->
    </transform>
    <models>
      <model file="sphere.3d" />
    </models>
  </group>
</group>
```

Figure 9: Excerto do ficheiro XML para o desenho da Terra e Lua

```
<!-- MARTE e LUAS-->
<group>
  <transform>
    <rotate angle="240" x="0" y="1" z="0"/> <!-- Rotação relativa ao Sol de Marte-->
    <translate x="39" y="0" z="0"/>
    <scale x="0.3" y="0.3" z="0.3"/> <!-- Tamanho de Marte -->
  </transform>
  <!-- MARTE -->
  <group>
    <models>
      <model file="sphere.3d" />
    </models>
  </group>
  <!-- PHOBOS -->
  <group>
    <transform>
      <translate x="0" y="1.2" z="1"/> <!-- Distância de Phobos a Marte -->
      <scale x="0.2" y="0.2" z="0.2"/> <!-- Tamanho de Phobos -->
    </transform>
    <models>
      <model file="sphere.3d" />
    </models>
  </group>
  <!-- DEIMOS -->
  <group>
    <transform>
      <translate x="0" y="-1" z="1.5"/> <!-- Distância de Deimos a Marte -->
      <scale x="0.1" y="0.1" z="0.1"/> <!-- Tamanho de Deimos -->
    </transform>
    <models>
      <model file="sphere.3d" />
    </models>
  </group>
</group>
```

Figure 10: Excerto do ficheiro XML para o desenho de Marte, Deimos e Fobos

Para desenhar o **anel de Saturno** foi aplicada a mesma estratégia dos dois subgrupos, um para Saturno e outro para o seu anel.


```

<!-- SATURNO -->
<group>
  <transform>
    <rotate angle ="30" x="0" y="1" z="0"/> <!-- Rotação relativa ao Sol de Saturno -->
    <translate x="66" y="0" z="0"/>
  </transform>
  <group>
    <transform>
      <scale x="5" y="5" z="5"/> <!-- Tamanho de Saturno -->
    </transform>
    <models>
      <model file="sphere.3d" />
    </models>
  </group>
  <group>
    <transform>
      <rotate angle ="20" x="1" y="0" z="1"/> <!-- Ângulo de rotação do anel de Saturno -->
      <scale x="1" y="1" z="1"/> <!-- Tamanho do Anel -->
    </transform>
    <models>
      <model file="torus.3d" /> <!-- Anel de Saturno -->
    </models>
  </group>
</group>

```

Figure 11: Excerto do ficheiro XML para o desenho de Saturno

Resultado para Sistema Solar

Na opção GL_LINE, o ficheiro XML criado produziu, através do *engine*, o seguinte resultado:

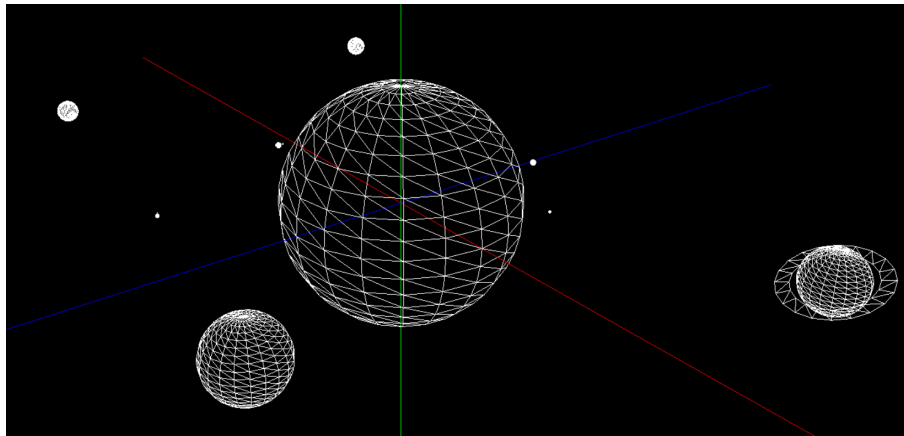


Figure 12: Sistema Solar

De modo a podermos verificar as distâncias relativas entre os planetas, foi também elaborado um ficheiro XML com os planetas do Sistema Solar alinhados no eixo do x (sem a presença das rotações), cujo resultado se encontra a seguir:

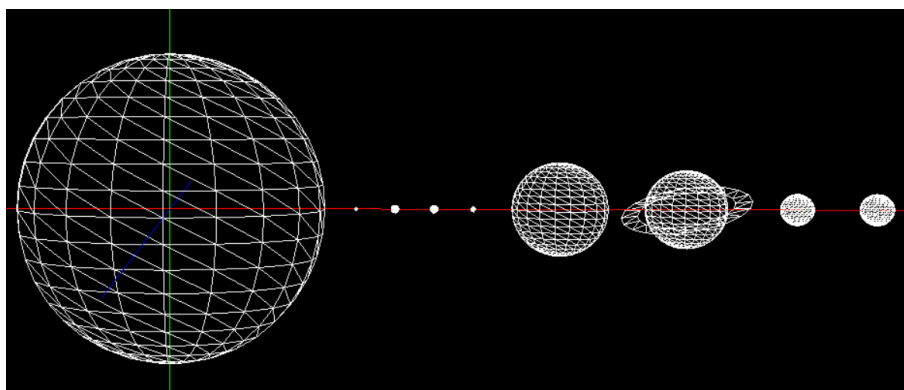


Figure 13: Sistema Solar alinhado

Como foi mencionado previamente, foram também acrescentadas os satélites naturais à Terra e a Marte. Deste modo, a seguinte imagem permite visualizar melhor esses astros, visto que não é fácil vê-las nas figuras acima devido à sua reduzida dimensão relativamente aos restantes astros.

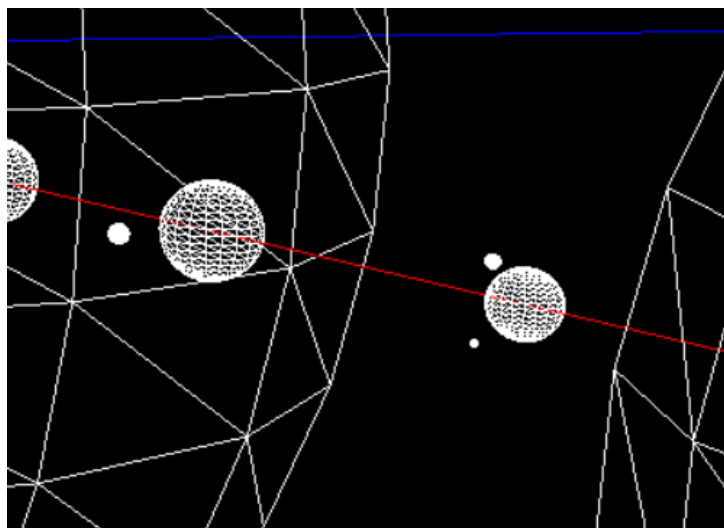


Figure 14: Terra e Marte com os seus satélites naturais

Extra (Casas)

Para complementar o nosso trabalho, decidimos gerar uma nova composição, sendo esta duas casas. Na fase em que nos encontramos limitámo-nos a esboçar superficialmente as casas. Detalhes como portas e janelas pretende-se que sejam inseridos posteriormente, uma vez que pretendemos utilizar texturas para o fazer.

Para modelar as casas (a base, os telhados e as chaminés) decidimos criar três novas figuras, as quais denominamos de *pyramid*, *roof* e *parallelepiped*.

A **pyramid** representa uma pirâmide e para criá-la são passados 3 argumentos, sendo estes o comprimento, a altura e a largura. A base da pirâmide é idêntico ao ilustrado na Figure 4. Unindo o ponto central da pirâmide a cada par de pontos da base, constróem-se as laterais da mesma.

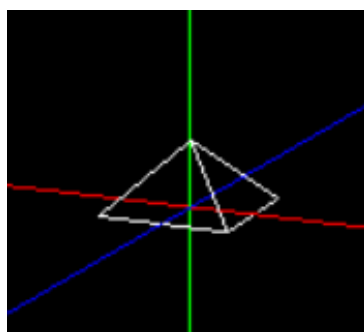


Figure 15: Desenho de uma pirâmide

Para desenhar o **roof**, também é necessário fornecer o comprimento, a altura e a largura para que desenhe algo semelhante à seguinte figura:

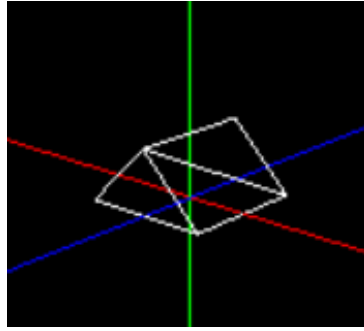


Figure 16: Desenho de um roof

Para representar esta figura, desenharam-se dois triângulos com uma distância no eixo dos z que corresponde à largura passada como argumento. Esses dois triângulos são ligados por retângulos, cada um formado por dois triângulos, como exemplificado na Figure 4. A cada duas lateral correspondentes dos primeiros triângulos, desenha um retângulo.

Para criar o **paralelepípedo** é necessário fornecer igualmente o comprimento, a altura e a largura. Com base nestes parâmetros, são desenhadas 6 faces recorrendo ao princípio ilustrado na Figure 4.

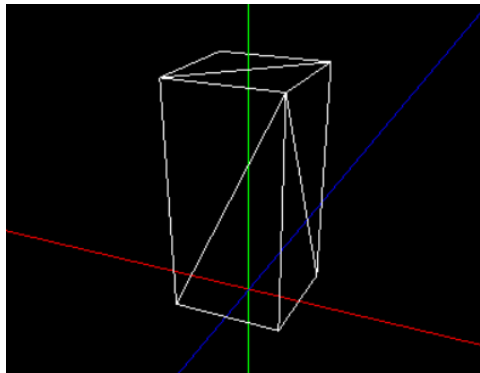


Figure 17: Desenho de um paralelepípedo

Finalmente, para desenhar as casas, foi então necessário criar um novo ficheiro XML. Este ficheiro chama-se teste_house.xml. Ele começa por configurar a câmara para que esteja numa posição que consideramos oportuna para visualizar as casas.

```
<window width="512" height="512" />
<camera>
  <position x="5" y="6" z="15" />
  <lookAt x="0" y="0" z="0" />
  <up x="0" y="1" z="0" />
  <projection fov="60" near="1" far="1000" />
</camera>
```

Figure 18: Configuração inicial da câmara

Dentro da tag *group* existem dois subgrupos, cada um correspondente a uma casa. Começamos por desenhar a base da casa com uso ao *paralelepípedo*, fazendo uma translação ao mesmo para a posição em x corresponda à que queremos desenhar todos os elementos da casa. Em seguida, dentro de cada um dos subgrupos, existem ainda 2 subgrupos, um correspondente ao telhado e outro à chaminé, com as suas respectivas transformações geométricas. Para desenhar a chaminé usou-se novamente a primitiva *paralelepípedo* e para desenhar os telhados usou-se para uma casa a primitiva *roof* e para a outra a *pyramid*.

```

<group>
  <transform>
    <translate x="-4" y="0" z="0" />
  </transform>
  <models>
    <model file="paralelepiped_4_4_4.3d" /> <!-- generator paralelepiped 4 4 4 paralelepiped_4_4_4.3d -->
  </models>
</group>
<group>
  <transform>
    <translate x="0" y="4" z="0" />
  </transform>
  <models>
    <model file="roof_4_2_4.3d" /> <!-- generator roof 4 2 4 roof_4_2_4.3d -->
  </models>
</group>
<group>
  <transform>
    <scale x="0.8" y="1" z="0.8" />
    <translate x="1.25" y="0.25" z="0" />
  </transform>
  <models>
    <model file="paralelepiped_1_2_1.3d" /> <!-- generator paralelepiped 1 2 1 paralelepiped_1_2_1.3d -->
  </models>
</group>
</group>

```

Figure 19: Excerto XML de uma casa

Relativamente à chaminé, optamos por desenhá-la afundada no telhado uma vez que esta abordagem seria mais fácil, levaria a menos erros de cálculo e faria com que possíveis problemas de passagem de luz não ocorram. Além disso, a parte enterrada da chaminé permanece oculta, de modo que a sua forma, sendo um paralelepípedo, não seja perceptível.

O resultado final é apresentado na seguinte figura.

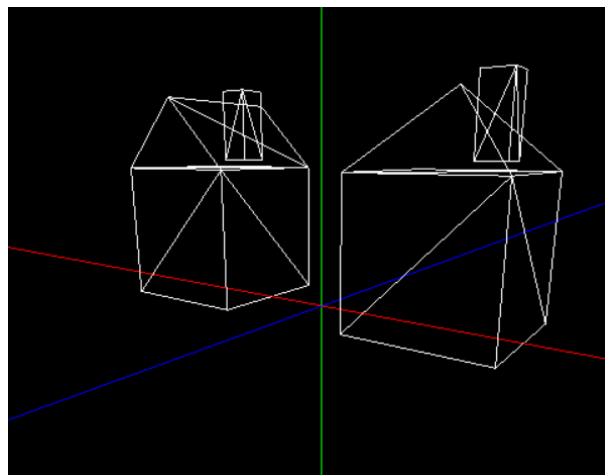


Figure 20: Resultado final produzido pelo nosso XML

Conclusão

Durante a segunda fase do trabalho, conseguimos de consolidar os diversos conhecimentos adquiridos relativos a transformações geométricas aplicados a modelos gerados.

Consideramos que fomos capazes de alcançar todos os objetivos propostos para esta fase, bem como alguns extras como a implementação da movimentação da câmara, a capacidade de alterar o modo de visualização da figura e a geração da primitiva do Torus, adicionada para aprimorar a representação de Saturno da realidade do Sistema Solar, atribuindo-lhe o seu anel. Acrescentamos, paralelamente, primitivas, nomeadamente uma pirâmide quadrada, um prisma triangular e um paralelepípedo, e um ficheiro xml para gerar casas que futuramente pretendemos complementar e melhorar durante as etapas seguintes.

Portanto, ao longo do desenvolvimento desta fase, consideramos que consolidamos os conceitos necessários, e como resultado, o grupo sente-se preparado para progredir para as próximas fases do projeto.