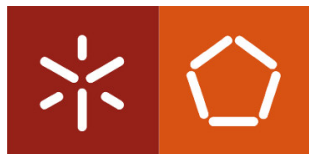


UNIVERSIDADE DO MINHO

ESCOLA DE ENGENHARIA

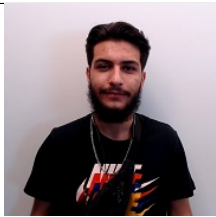





Computação Gráfica

Licenciatura em Engenharia Informática

Computação Gráfica - Fase II

Grupo 39

A93221	A93256	A93169	A88539
			
André Vaz	João Mendes	Laura Rodrigues	Luís Fernandes

Abril 2022

Índice

1	Introdução	3
2	Estrutura do ficheiro XML	4
3	Engine	5
3.1	World	5
3.2	Group	5
3.3	Transforms	6
3.4	Render	6
4	*Bonus* Teclado	6
5	Testes	7
6	Sistema Solar	8
7	Conclusão	10

Lista de Figuras

1	Exemplo de formatação de XML	4
2	Testes exemplificativos	7
3	Sistema solar	8
4	Sistema solar com rotações ilustrativas	8
5	Sistema solar com foguetão e wormhole	9
6	Close-up do Foguetão	9

1 Introdução

A segunda fase do trabalho prático teve como objetivo a criação de cenários hierárquicos usando transformações geométricas. Apenas o Engine foi alterado de modo a conseguir responder à criação de um cenário gráfico em 3D que representasse o Sistema Solar, recorrendo à leitura e interpretação de um ficheiro XML e do usando primitivas gráficas para representar as diferentes formas geométricas.

Para além do objetivo principal desta fase, ou seja, as transformações geométricas de translação, rotação e escala foi também implementada a cor. De modo a facilitar a visualização de todos os cenários criados foi implementada a rotação do cenário.

2 Estrutura do ficheiro XML

Para além das informações referentes à fase anterior, ou seja, a câmara, qualquer ficheiro XML usado para representar uma cena, tem na sua essência uma estrutura em árvore, onde cada nodo contém um conjunto de transformações geométricas e/ou um conjunto de modelos. Para além disso, esta especificado a camera.

Eis um exemplo de um ficheiro XML:

```
<world>
  <camera>
    <position x="10" y="3" 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>
    <transform>
      <translate x="0" y="1" z="0" />
    </transform>
    <models>
      <model file="box.3d" />
    </models>
  </group>
</world>
```

Figura 1: Exemplo de formatação de XML

Como podemos verificar na figura acima, cada nodo corresponde a um *group* que é definido por um conjunto de elementos, podendo estes ser transformações, modelos ou subgrupos.

- **Elemento *Rotate***: refere-se à rotação de um objeto por ângulo e um vetor rotação definido por x, y e z.
- **Elemento *Translate***: refere-se à translação de um objeto por um vetor definido por x, y e z.
- **Elemento *Scale***: refere-se à escala de um objeto através de escalares x, y e z.
- **Elemento *models***: indica quais os modelos *.3d* a desenhar
- **Elemento *color***: indica a cor atribuída a esses objetos

3 Engine

Relativamente ao *Engine* foi continuado o que já tinha sido feito na fase anterior sendo apenas necessário acrescentar a nova funcionalidade das transformações.

3.1 World

Esta classe foi alterada passando a conter apenas uma *Camera* (desenvolvida na fase anterior) e um *Grupo*.

Ademais, o *parse* do ficheiro *XML* foi enriquecido, através da adição do *parseTransform* e ainda o aperfeiçoamento do *parseGroup*.

3.2 Group

O grupo é definido da seguinte forma:

- ***vector<models>*** conjunto de modelos do grupo.
- ***vector<translate>*** conjunto de *translates* a serem realizados no grupo e seus subgrupos.
- ***vector<rotate>*** conjunto de *rotates* a serem realizados no grupo e seus subgrupos.
- ***vector<scale>*** conjunto de *scales* a serem realizados no grupo e seus subgrupos.
- ***color*** cor a ser usada para desenhar os modelos do grupo.
- ***vector<group>*** conjunto de *subgrupos* do grupo.

Como o ficheiro *XML* contém um grupo principal o mesmo vai ser lido através da função *parseGroup*, que recebe como argumentos o *path* onde os *models* estão guardados, o elemento *XML* do grupo a ser lido e a classe *Group* para onde o mesmo vai ser carregado.

Todos os seus elementos filho serão carregados para a sua estrutura correta para mais tarde serem desenhados. Caso encontre outro elemento do tipo *group* a função será executada recursivamente, será criado um novo *Group* para ser passado como argumento na função e adicionado à lista de subgrupos do grupo que estava a ser lido previamente.

3.3 Transforms

Foram criadas várias classes com o intuito de guardar as transformações a serem realizadas como *Rotate*, *Translate*, *Scale* e *Color*. Para além disso, elaborámos para cada transformação os *getters* e *setters* das suas variáveis, bem como os seus respetivos construtores.

Rotate conjunto de 3 *floats*: x, y e z.

Translate conjunto de 4 *floats*: angle, x, y e z.

Scale conjunto de 3 *floats*: x, y e z.

Color conjunto de 3 *floats*: r, g e b.

As transformações pertencem ao elemento *transforms* que por sua vez pertence ao elemento *group*. Quando este for lido no *parseGroup* a função *parseTransforms* será chamada para carregar os elementos de *transforms* para as suas estruturas corretas no *group* em que foi detetado.

3.4 Render

De forma a aplicar as novas estruturas criadas, a maneira como os modelos são renderizados na função *renderModels* foi alterada, passando a dar *push* de uma cópia da matriz anterior para serem aplicadas as transformações do grupo, usando a função *transformacoes*. De seguida, os modelos do grupo são desenhados usando a cor do grupo, que é passada como argumento para a função *drawModel*.

Com os modelos desenhados, a função será chamada recursivamente para os diferentes sub-grupos, sendo no fim realizado um *pop* da *stack* de matrizes, assim as transformações que são passadas de grupo-pai para grupo-filho, mas não de grupo-filho para grupo-pai.

4 *Bonus* Teclado

Foi adicionada a funcionalidade de teclado que permite interagir com o cenário a ser apresentado. Assim, é agora possível rodar o **World** que é apresentado. As teclas reponsáveis são:

- - rodar camera para a direita
- ← - rodar camera para a esquerda
- f - ligar/desligar eixos

5 Testes

Testes referentes à segunda fase fornecidos pela equipa de docentes, com cor adicionada.

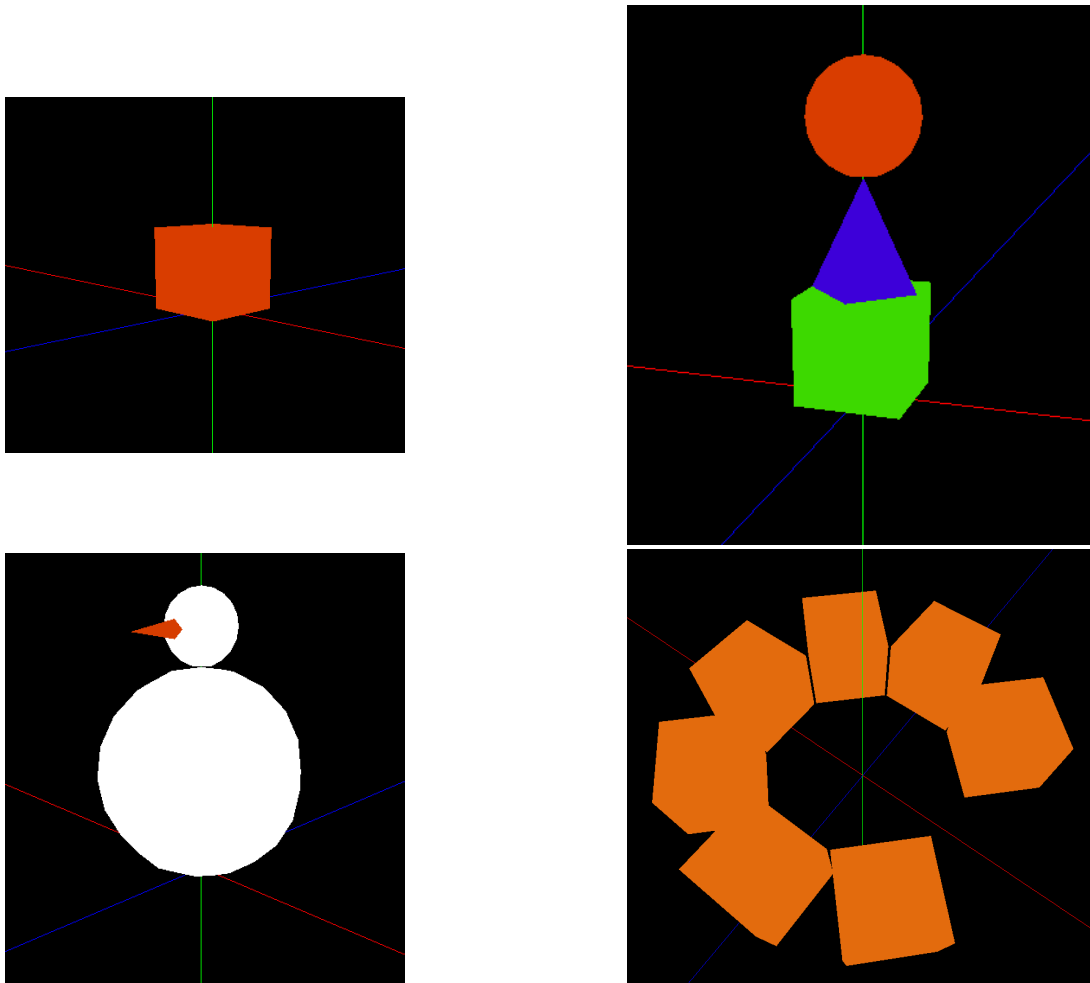


Figura 2: Testes exemplificativos

6 Sistema Solar

Nesta segunda fase, foi-nos pedida uma demonstração estática do modelo do sistema solar, que contém o sol e os vários planetas e as respetivas luas, através de um ficheiro XML personalizado.

Para a realização deste modelo tivemos em consideração as dimensões relativas de cada planeta de forma a criar uma representação realista dos sistema solar.

O ficheiro XML desenvolvido está organizado conforme a distância dos planetas relativamente ao sol, começando pelo sol e avançando até neptuno. Para além disso, foram também representados alguns dos satélites naturais dos respetivos planetas.

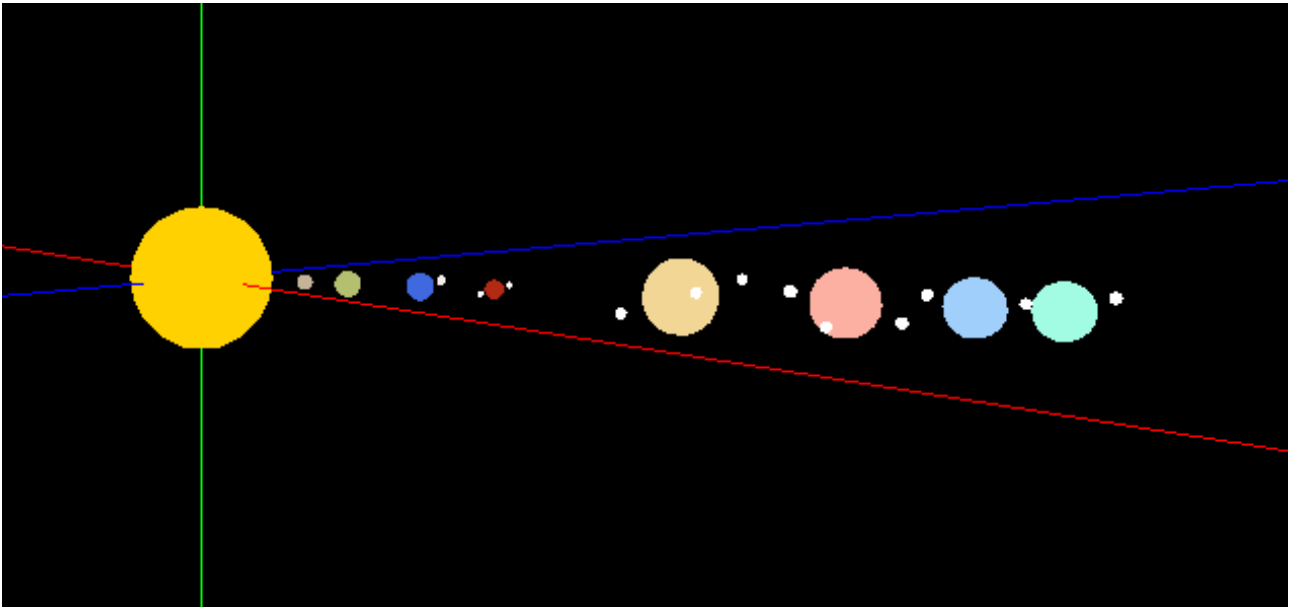


Figura 3: Sistema solar

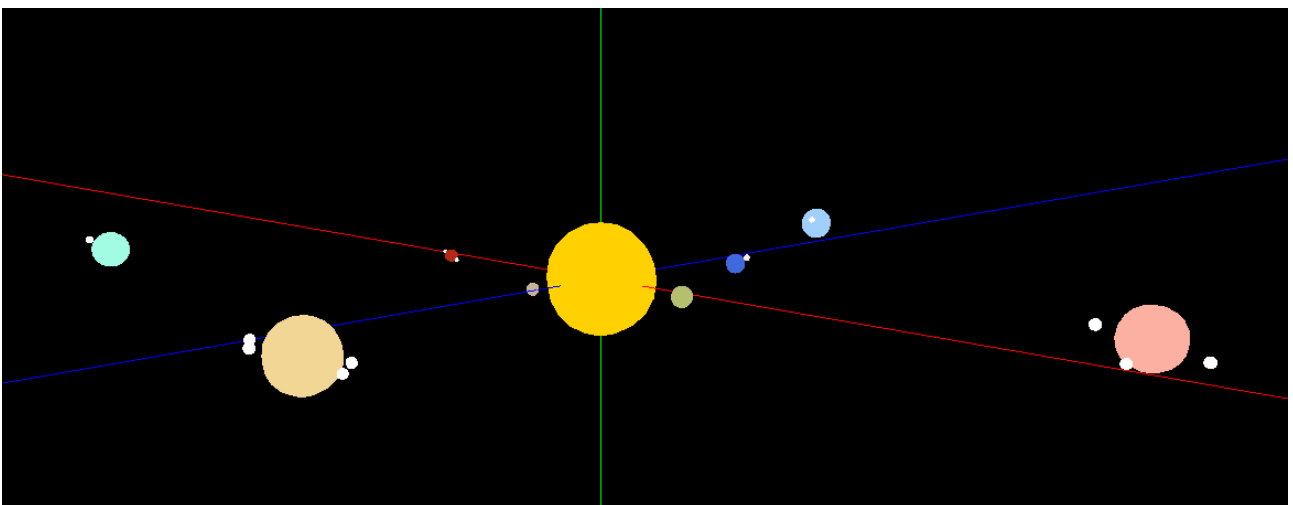


Figura 4: Sistema solar com rotações ilustrativas

Adicionalmente foram desenvolvidas duas estruturas adicionais: um Wormhole e um foguetão. O primeiro é constituído por um tronco de grande amplitude e por um cone, permitindo assim criar um efeito de funil.

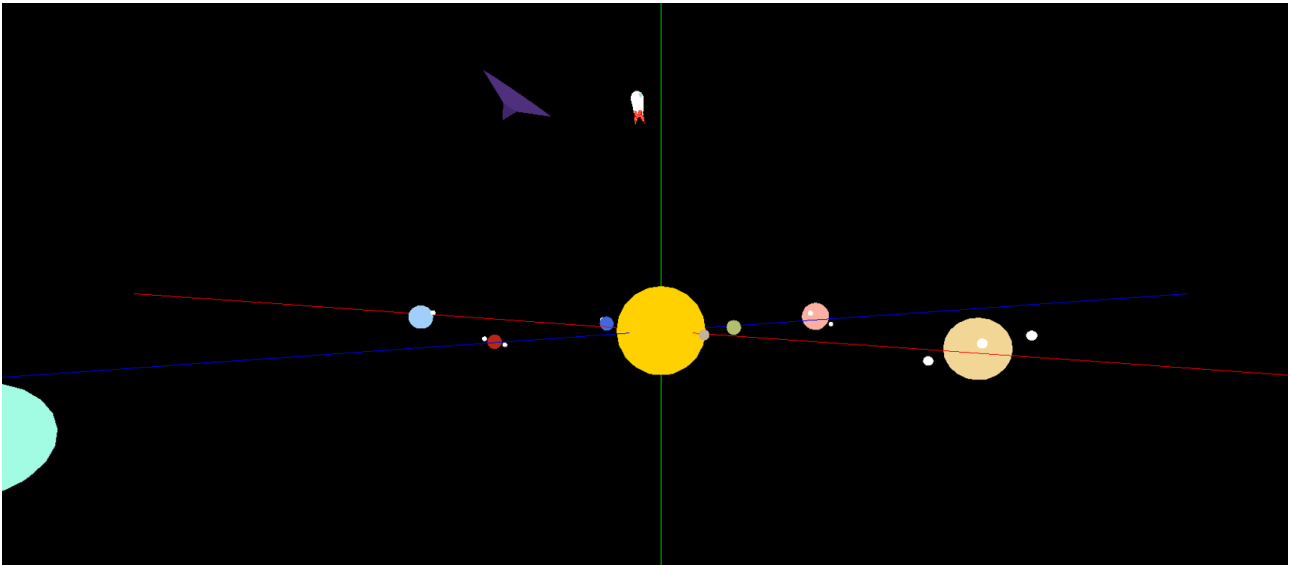


Figura 5: Sistema solar com foguetão e wormhole

Por sua vez, o foguetão é constituído por várias sobreposições de figuras. O corpo (a branco) foi obtido através de translações e é formado por um tronco de raio superior maior que o inferior e por uma esfera que se sobrepõe à face superior. Tem duas esferas mais pequenas a azul que representam as janelas. Por fim, os apoios são constituídos por quatro cones que sofreram rotações de modo a ficarem inclinados. A dupla tonalidade (laranja e vermelha) foi obtida através da sobreposição de dois cones de duas cores.

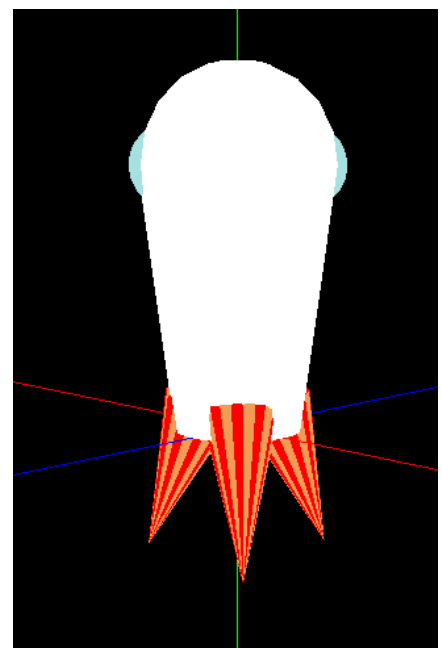
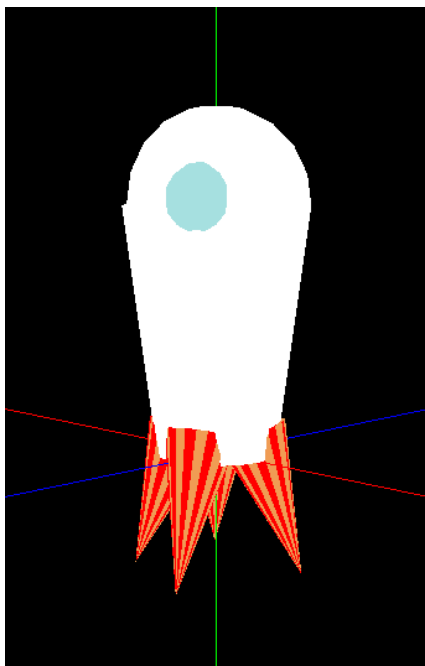


Figura 6: Close-up do Foguetão

7 Conclusão

O projeto teve como intuito melhorar o motor gráfico iniciado na fase anterior, através da especificação e interpretação de cenários hierárquicos em xml.

Nesta fase do projeto, foram adicionadas poucas estruturas novas uma vez que grande parte já tinha sido definida anteriormente, na primeira fase.

Posto isto, houve o cuidado de cumprir todos os parâmetros pedidos bem como elaborar requisitos extras, como a implementação das cores de forma a tornar mais perceptível a diversidade de planetas e satélites. Para além disso, foi implementada a rotação da câmara e a opção de ativar/desativar o eixo *xyz*.

Por fim, consideramos que houve um balanço positivo, uma vez que ultrapassámos com sucesso as dificuldades sentidas na elaboração, acabando por cumprir com todos os requisitos pedidos.