



Universidade do Minho

Escola de Engenharia

Mestrado Integrado em Engenharia Informática

Computação Gráfica

Ano Letivo de 2016/2017

Fase 2 – Transformações Geométricas

A27748 - Gustavo José Afonso Andrez

A74634 - Rogério Gomes Lopes Moreira

A76507 - Samuel Gonçalves Ferreira

A71835 – Tiago Filipe Oliveira Sá

Índice

Índice	2
Introdução	3
Descrição da Estrutura de Dados	4
Descrição do Processo de Leitura	5
Descrição do Ciclo de Rendering	6
Descrição do Ficheiro XML – Sistema Solar	7
Conclusão	8

Introdução

O trabalho prático apresentado, do qual este relatório diz apenas respeito à Fase 2, tem como objetivo desenvolver competências na área dos gráficos 3D e no desenvolvimento de um motor para tal. O trabalho está dividido em 4 fases.

Esta segunda etapa é constituída essencialmente por duas partes:

- **Drawer** – implementação de transformações geométricas (translação, rotação e escala) no drawer, já desenvolvido na fase 1 da leitura.
- **Planeta Solar (XML)** – ficheiro em XML contendo a informação do sistema solar.

As aplicações foram desenvolvidas recorrendo ao Visual Studio e à linguagem de programação C++.

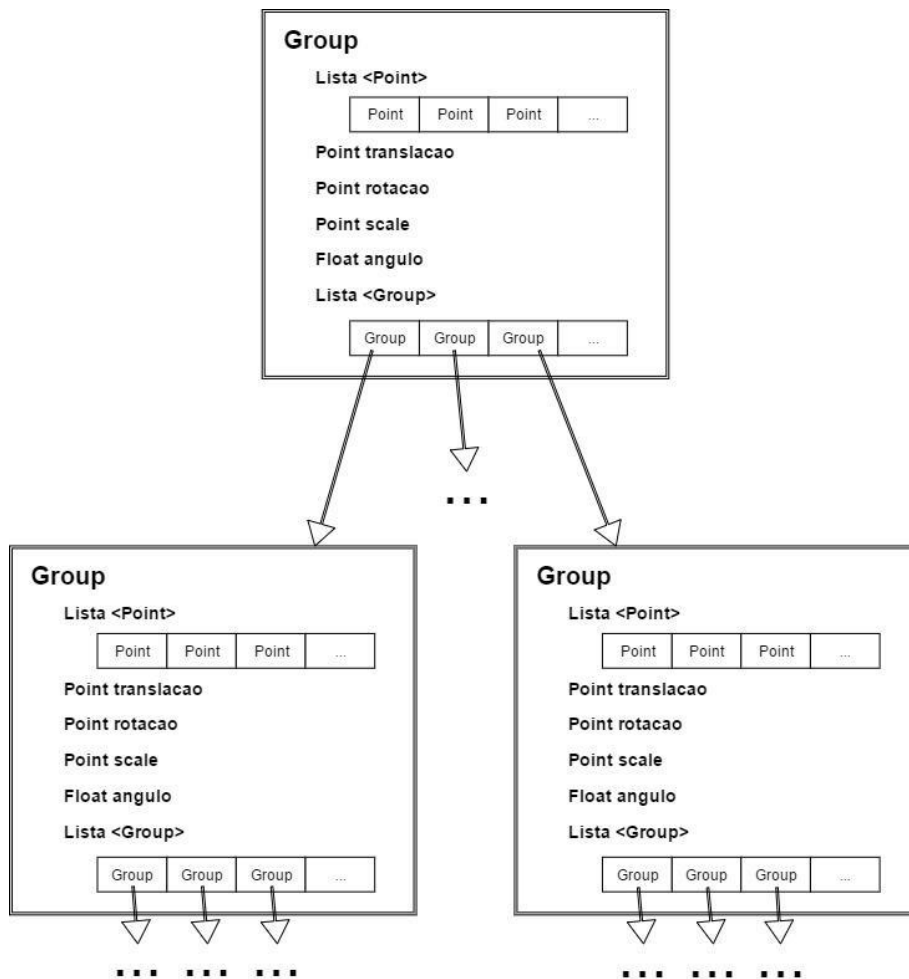
Descrição da Estrutura de Dados

O primeiro passo na resolução do problema foi criar uma estrutura de dados capaz de armazenar toda a informação contida no ficheiro XML (e nos ficheiros por este apontados) de maneira a que a leitura do ficheiro seja apenas efetuada uma vez e que seja possível iterar sobre ela de forma a que cada grupo herde as transformações geométricas de todos os grupos dos quais ele descende.

Para satisfazer estas necessidades, foi então criada como variável global uma lista de grupos (`vector<Group> scene`), sendo a classe `Group` definida da seguinte maneira:

```
class Group {
public:
    vector<Point> points;
    Point translation;
    Point rotation;
    Point scale;
    float angle;
    vector<Group> childs;
};
```

A estrutura `Group` descrita é esquematizada na figura seguinte:



Descrição do Processo de Leitura

Depois que construída a estrutura de dados, tem que ser efetuada a leitura do ficheiro XML que contém toda a informação da cena a desenhar, neste caso o Sistema Solar.

- O primeiro passo é tentar abrir o ficheiro passado como argumento ao drawer;
- Caso o ficheiro tenha sido aberto com sucesso, é atribuído a uma variável o *root element* do mesmo, que no nosso caso é a tag *scene*. Dentro desta tag estão contidos todos os grupos de uma determinada cena;
- Cada grupo contido dentro da tag *scene*, do tipo *TiXmlElement*, é convertido

para um objeto do tipo *Group*, através da função *CalculateGroupPoints*, e posteriormente adicionado à lista de grupos *scene*.

A função *CalculateGroupPoints* recebe um elemento *TiXmlElement* e converte para um objeto do tipo *Group*, retornado no final da sua execução, objeto esse que contém a informação relativa ao grupo passado como argumento e a todos os seus grupos filho. Esta função é implementada da seguinte forma:

- Para cada grupo filho do grupo recebido como parâmetro é feita uma invocação recursiva cujo resultado é armazenado na lista *childs* do grupo a ser convertido;
- São inicializadas de seguida as variáveis de translação e rotação a 0 e as variáveis de escala a 1;
- Caso existam informações sobre transformações geométricas associadas ao grupo no ficheiro XML, estas serão armazenadas nas variáveis correspondentes (as que foram inicializadas anteriormente);
- Por fim, é feita a verificação se existem *models* associados ao *Group* em questão e em caso afirmativo vai percorrer todos esses models. Para cada um deles, abre o ficheiro *.3d* correspondente contendo os pontos da figura a ser renderizada e adiciona-os à lista *points* do grupo em questão.

Descrição do Ciclo de Rendering

A renderização de uma cena é feita através da invocação da função *drawScene* que, recebendo uma lista de objetos do tipo *Group*, desenha todos esses grupos com o seguinte algoritmo:

Para cada um dos *Group* recebidos como argumento:

- *glPushMatrix()* – Duplica a matriz *modelView* atual e adiciona-a ao topo da *stack* de matrizes que representam as transformações geométricas que vão ser consideradas aos pontos a ser desenhados.

- São aplicadas as transformações geométricas *translate*, *rotate* e *scale* à matriz *modelView*. Os parâmetros necessários para estas transformações geométricas são obtidos a partir da memória, da instância *Group* em questão. Recorde-se que a estrutura de dados responsável por armazenar um *Group* contém informações relativas aos pontos e ângulo das transformações geométricas.
- É feita de seguida uma invocação recursiva que vai desenhar todos os grupos filho do grupo atual, sendo estes também desenhados com as transformações geométricas presentes atualmente na matriz *modelView*.
- De seguida, é realizado um ciclo que itera sobre a lista de pontos do grupo e os desenha, três a três já que todas as figuras desenhadas são construídas à custa de triângulos.
- Por fim, para que as transformações geométricas deste grupo não sejam aplicadas a nenhum outro grupo a ser desenhado numa iteração seguinte usa-se a função `glPopMatrix()`, que elimina a matriz que se encontrar no topo da *stack* de matrizes.

Descrição do Ficheiro XML – Sistema Solar

O ficheiro XML possui os dados relativos ao sistema solar (planetas, estrela e luas) a ser representado. Este ficheiro encontra-se dividido em níveis da seguinte forma:

- Nível 1 - Consiste num grupo tem por função descrever as transformações geométricas (translação, escala e rotação) a aplicar a todas as figuras (estrela, planetas e luas) contidas no ficheiro.
- Nível 2 - Conjunto de grupos relativos às figuras correspondentes à estrela e planetas principais. Assim, para além das respetivas transformações geométricas de cada astro, existe informação acerca do ficheiro *.3d* correspondente. Este nível inclui os astros Sol, Mercúrio, Vénus, Terra, Marte, Júpiter, Saturno, Urano, Neptuno e Plutão.
- Nível 3 - Este nível corresponde ao subnível dos planetas que possuem luas, caso da Terra, Júpiter, Saturno e Úrano. Assim, dentro do grupo destes

planetas existe um ou mais grupos, cada um contendo os parâmetros respeitantes ao satélite natural. Este nível inclui um total de 14 astros relativos às luas dos planetas Terra, Júpiter, Saturno, Urano e Neptuno.

O ficheiro está delimitado pelas tags scene.

Conclusão

Com a elaboração desta fase do projeto concluímos acerca da forma de aplicação de transformações geométricas a modelos existentes e também acerca da organização de *scenes* em hierarquia de *groups*.

Relativamente ao trabalho desenvolvido deixamos algumas observações:

- o tamanho relativo entre os planetas e luas corresponde é próximo do real;
- apercebemo-nos da necessidade de respeitar a ordem de transformações;
- devido à diferença de tamanho dos vários astros, foram utilizados dois ficheiros de esferas para a sua geração de modo a adequar o número de *stacks* e *slices*;
- foi criado um segundo ficheiro XML com uma distribuição não colinear dos planetas por forma a criar um modelo mais realista.

Considerando o desenvolvimento do trabalho, consideramos que os objetivos desta fase do trabalho foram cumpridos.