

UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA
INFORMÁTICA

COMPUTAÇÃO GRÁFICA

Transformações Geométricas

Grupo:

Etienne Costa A76089

Joana Cruz A76270

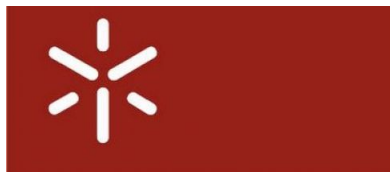
Rafael Alves A72629

Maurício Salgado A71407

Docente:

António Ramires

24 de Março de 2019



Conteúdo

1	Introdução	2
2	Leitura e Processamento de um ficheiro XML	2
3	Armazenamento de dados em classes	2
3.1	Vertex	2
3.2	Model	3
3.3	Group	3
4	Parser	4
5	Exemplos de Execução	4
5.1	Figuras primitivas	4
5.2	Figuras primitivas com diferentes escalas	4
5.3	Sistema solar estático	4
6	Conclusão	4

1 Introdução

O relatório apresentado diz respeito à segunda fase do projeto proposto no âmbito da unidade curricular de Computação Gráfica. O trabalho consiste na criação de um cenário através do parsing de ficheiros XML e aplicação de várias transformações geométricas hierquicamente em OpenGL tal como translações, rotações e escalas.

2 Leitura e Processamento de um ficheiro XML

O processamento de um cenário em formato XML pode ser visto como duas fases distintas:

- Leitura e Parsing do cenário – Consiste na abertura do ficheiro que contém o cenário que em modo de leitura e extração da respetiva hierarquia em XML. Nesta fase são também retiradas as componentes que caracterizam uma transformação geométrica (translação, escala ou rotação) ou desenho de uma primitiva.
- Armazenamento nas estruturas de dados – De modo a se conseguir redesenhar um modelo quantas vezes for necessário, as instruções que caracterizam uma transformação geométrica são armazenadas em estruturas adequadas às mesmas.

Para uma melhor organização do código, optou-se por separar estas duas fases do programa Engine num conjunto de módulos apropriados às mesmas. As duas subsecções seguintes descrevem com detalhe a organização desses módulos.

3 Armazenamento de dados em classes

3.1 Vertex

Esta classe representa um ponto num referencial a três dimensões, com as coordenadas x, y e z, o que se torna bastante útil para a representação dos vértices utilizados posteriormente para o desenho dos triângulos que elaboram as figuras primitivas.

```
class Vertex{
    public:
        float x;
        float y;
        float z;
        Vertex();
        Vertex(float xx, float yy, float zz);
```

```

~Vertex();
};

```

3.2 Model

Armazena todos os pontos para a criação das figuras primitivas (plane, box, cone, sphere e torus) estão presentes e descritos neste ficheiro, assim como a cor escolhida para o modelo.

```

class Model{
public:
    string fileName;
    vector<Vertex> vertexes;
    Vertex color;
    Model();
    Model(string path, Vertex color);
    ~Model();
};

```

3.3 Group

Tal como sugere o nome, esta é a classe principal de armazenamento que guarda os dados retirados do ficheiro XML. Nesta estrutura é possível armazenar todas as informações que estão associadas a uma determinada figura tal como as transformações. A figura, por sua vez, é representada por um ficheiro com extensão 3d onde se pode encontrar todos os pontos que a constitui. Sendo assim, achou-se conveniente armazenar igualmente o nome do ficheiro e todos os vértices contidos neste.

```

class Group{
public:
    Vertex rotation;
    float rotationAngle;
    Vertex translation;
    Vertex scale;
    vector<Model> models;
    vector<Group> subGroups;
    Group(void);
    Group(Vertex rotation, float rotAngle, Vertex translation,
    Vertex scale, vector<Model> models, vector<Group> subGroups);
    ~Group();
};

```

4 Parser

Esta classe é fundamento para o bom funcionamento do engine porque é este que efetua o parsing do ficheiro XML. Desta forma, o Parser é responsável por inserir toda a informação encontrada no documento XML num vetor de Struct.

- ParseXMLFile - sendo o cenário sempre um conjunto de grupos, esta função verifica se o documento XML apresenta um formato correto e após isso vai começar a efetuar o parsing de cada group;
- ParseGroup percorre um group do XML e extrai a informação correspondente. Se o grupo em questão contiver outros grupos dentro de si, essa informação também será processada. Caso a informação a ler seja uma transformação a função parseAttributes é invocada;
- ParseModel processa a informação correspondente ao ficheiro da figura primitiva(vértices) e a cor;
- ParseAttributes que processa a informação sempre que é encontrada uma transformação (translate, rotate ou scale) criando um vértice e adicionando-o ao Group que está a ser elaborada pelo parseGroup.

5 Exemplos de Execução

5.1 Figuras primitivas

5.2 Figuras primitivas com diferentes escalas

5.3 Sistema solar estático

6 Conclusão