

# Relatório

## Fase 2

*5-Abril-2024*  
*Computação Gráfica*

*Luís Ribeiro, A100608*  
*Martim Félix, A100647*  
*Diogo Santos, A100714*

# INTRODUÇÃO

Nesta segunda fase do projeto de Computação Gráfica, é pedido para implementar a noção de grupo nas “cenas”.

Estes grupos definem uma hierarquia de transformações geométricas e modelos aos quais estas transformações se aplicam. Cada grupo pode também ser e ter subgrupos. Esta relação estabelece que um grupo B definido dentro dos limites de um grupo A também sofre as suas transformações. Os modelos definidos num dado grupo sofrem o conjunto de transformações observáveis no “caminho” até eles.

É também pedido para criarmos uma demo simples do sistema solar. Para isso basta-nos definir o ficheiro *xml* que possa definir a “cena”. A componente implementada nesta 3a fase é útil para estabelecer as relações de órbita. Sabemos que a Terra orbita em torno do Sol e que a Lua, por sua vez, orbita em torno da Terra assim, na prática, podemos observar algo como:

```
grupo{
  Sol,
  grupo{
    Mercurio,
    Venus,
    grupo{
      Terra,
      grupo{
        Lua
      }
    },
    Marte,
    ...
  }
```

Além dos dois requisitos referidos acima implementamos também, nesta fase outras funcionalidades *extra*. Funcionalidades como um elemento *xml* “color” utilizada para “pintar” um determinado grupo com uma determinada cor que achamos acrescentar bastante valor à nossa demo de sistema solar. Acrescentamos também atalhos do teclado que nos permitem dar *toggle* aos eixos e ao tipo de desenho dos modelos. Desenho preenchido ou demarcando apenas as arestas do triângulo.

# ENGINE

Para completar os objetivos propostos para esta segunda fase foram necessárias algumas alterações à funcionalidade do componente *Engine*.

Em primeiro lugar foi necessário alterar o *parser* para conseguir interpretar corretamente os novos parâmetros presentes no *xml*. Depois da aquisição de dados seria necessário representar os dados obtidos numa estrutura que permitisse representar corretamente a hierarquia presente nas componentes *grupo* do ficheiro *xml* de entrada. A partir desta estrutura, “recuperamos” os dados e desenhámos a “cena”.

A estrutura escolhida para concretizar então estes grupos foram *Rose Trees* uma vez que a noção de hierarquia era preservada pela travessia da mesma e o requisito de um grupo ter *n* sub-grupos também ficava garantido pelo número *n* de sub-nodos de um nodo de uma *Rose Tree*.

Concretizando então esta visão o resultado do parse do ficheiro resulta na seguinte estrutura:

## Estrutura de Dados:

World

```
{
    Int WindowWidth
    Int WindowHeight
    Vector<Point> lookAt;
    Vector<float> projection;
    GroupNode* rootGroup;
}
```

GroupNode

```
{
    Vector<GroupNode * sub_nodes;
    Vector<Transform *> transforms;
    Vector<string *> models;
}
```

## Desenho a partir da estrutura auxiliar:

A partir da estrutura acima, definida com recurso ao ficheiro *xml* de entrada procedemos então ao desenho da “cena”.

Além do processo que existia da fase anterior de definir os parâmetros do *gl*, introduz-se o desenho de “grupos”. Para isso utilizamos o método *draw* sobre o *GroupNode* raiz presente no objeto *World* criado. Este método está definido assim:

```
void GroupNode::draw() {
    glPushMatrix();

    for (const Transform *t : transforms) {
        t->applyTransform();
    }

    glColor3f(color[0], color[1], color[2]);

    drawModels(parse3dFiles(models));

    for (GroupNode * node : sub_nodes) {
        node->draw();
    }

    glPopMatrix();
}
```

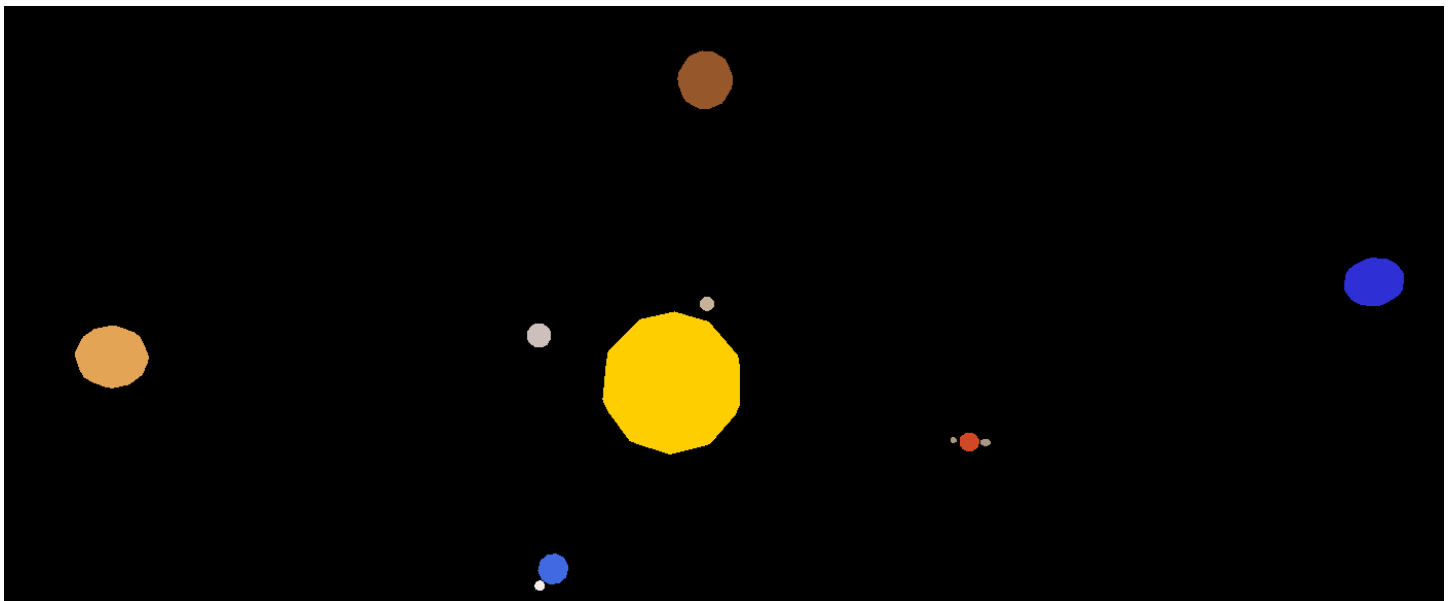
Em primeiro lugar recorreremos à *stack* de matrizes para preservar a matriz recebida, depois iteramos sobre as iterações definidas no nosso grupo e aplicamo-las. De seguida “pintamos” os objetos do grupo com a cor lida. Depois disso desenhamos os modelos, recorrendo antes ao *parse* dos ficheiros guardados no vetor *models*. Finalizamos com uma chamada recursiva sobre os sub-nodes. É importante lembrar que uma vez que o *popMatrix()* só é chamado depois destas chamadas recursivas, estas ainda sofrem as transformações aplicadas no grupo ativo, como se pretendia.

Em relação ao desenho dos modelos, o processo mantém-se exatamente o mesmo comparativamente à fase anterior. Dado o *parse* dos pontos compõe-se o *VBO* e desenha-se o modelo.

# Sistema Solar

De forma a cumprir com o propósito desta fase do trabalho, implementamos também um modelo estático do sistema solar. Este modelo tenta ter em consideração as dimensões relativas de cada planeta, mas o objetivo principal foi conseguir uma representação agradável do que seria possível no nosso *engine*.

Colocamos assim em anexo uma imagem do sistema solar apresentado:



*Figura 1: Sistema Solar*

# Conclusão:

Dada por concluída a fase 2 do projeto, consideramos interessante tomar algumas considerações sobre o trabalho desenvolvido.

No espectro positivo, consideramos conveniente destacar o correto funcionamento do nosso programa. Além disso, as estruturas implementadas estão em concordância com a estrutura do XML o que permite uma visualização mais clara daquilo que está a ser armazenado.

Numa luz menos positiva concluímos que esta fase era uma boa oportunidade para “caprichar” em mais *features* extra, mas em última instância investimos mais numa implementação sólida da fase 2 que nos permitisse subir mais facilmente a escada da complexidade nas fases seguintes.

Estamos satisfeitos com o este trecho de desenvolvimento e confiantes para as próximas fases.