Universidade do Minho
Licenciatura em Engenharia Informática

Unidade Curricular de Computação Gráfica

Ano Letivo de 2024/2025

Practical Assignment CG – Fase 2

Adélio Fernandes	A78778
Tiago Carneiro	A93207
Tiago Guedes	A97369

29 de março de 2025

CG

Índice

1. Introdução

No âmbito da Unidade Curricular de Computação Gráfica (CG), foi desenvolvido um projeto dividido em várias fases, sendo este relatório referente à segunda fase. Esta etapa tem como principal objetivo a introdução de transformações geométricas para a criação de cenas hierárquicas, permitindo a organização de modelos 3D em estruturas em árvore.

Assim como na fase anterior, esta etapa do projeto pode ser dividida em duas partes principais. A primeira foca-se no motor gráfico, tornando-o capaz de interpretar e aplicar transformações geométricas em objetos e grupos de objetos de forma hierárquica. O motor deve ser capaz de processar ficheiros XML que definem a cena, interpretando transformações como translação, rotação e escala.

A segunda parte diz respeito à construção de um ficheiro XML que define um modelo estático do Sistema Solar. Nesta estrutura, os elementos (Sol, planetas e luas) são organizados hierarquicamente.

A hierarquia de transformações geométricas é um aspeto crucial desta fase, uma vez que cada cena pode conter um número arbitrário de sub-cenas, com cada nível herdando as modificações realizadas no nível imediatamente anterior. Assim, a ordem pela qual as transformações são aplicadas tem um impacto direto no resultado final.

2. Alterações e melhorias

Nesta segunda fase do projeto, o motor gráfico foi ampliado para suportar cenas hierárquicas, introduzindo transformações geométricas compostas (translação, rotação, escala) aplicadas a grupos de modelos. Enquanto na Fase 1 tínhamos modelos carregados diretamente num vetor global e renderizados sem hierarquia, na Fase 2 adicionamos uma estrutura de árvore onde cada grupo (classe group) contém as suas próprias transformações, modelos e subgrupos. Esta mudança permite que objetos como o sistema solar sejam representados com relações pai-filho, onde as transformações de um grupo afetam todos.

Para isso atualizamos a classe Figure da fase1 que era responsável por representar a geometria dos modelos 3D, armazenando vértices, normais e triângulos, além de carregar dados de arquivos externos (como cone.3d) para passar a se integrar a um sistema hierárquico de cena. Esta mudança foi crucial para suportar a hierarquia de cena, onde um mesmo modelo (ex.: um planeta) pode aparecer em diferentes posições e escalas, dependendo do grupo pai.

Para isso introduzimos a noção de grupos (classe Group), que não existia na Fase 1. Enquanto na Fase 1 todos os modelos eram renderizados em um único contexto espacial, agora os modelos podem ser organizados em árvores, com transformações cumulativas herdadas dos pais. A classe Figure, portanto, deixou de ser usada isoladamente e passou a ser encapsulada dentro de grupos, ganhando maior flexibilidade.

No parser XML passamos a interpretar a estrutura hierárquica lendo recursivamente subgrupos e transformações começando a aplicar transformações acumuladas hierarquicamente, substituindo a abordagem estática da Fase 1.

3. Atualizações ao motor

3.1. Parser XML

A estrutura do projeto foi redesenhada para acomodar a complexidade das cenas hierárquicas levando a que fossem necessárias mudanças no parser de XML.

Na Fase 1, o parser de XML lia apenas modelos simples e configurações básicas da câmera e janela. Na Fase 2, o parser foi expandido para interpretar a estrutura hierárquica de grupos definida no XML. Agora, a função parseGroup processa recursivamente cada <group>, identificando transformações (<translate>, <rotate>, <scale>), modelos (<model file="...">) e subgrupos aninhados. Cada transformação é armazenada em objetos das classes Translate, Rotate ou Scale, que são associados ao grupo correspondente.

Um avanço significativo foi a capacidade de carregar modelos diretamente nos grupos, em vez de usar um vetor global.

3.2. Reorganização dos módulos

Devido a todas as alterações necessárias para abranger a nova e melhorada estratégia de parsing, foi então decidido proceder a uma reorganização dos módulos desenvolvidos até ao momento, de modo a facilitar não só a procura e leitura dos mesmos, como também a sua modularidade e escalabilidade para futuras fases de desenvolvimento.

Foi então criada a classe Transform que contém as transformações geométricas que podem ser aplicadas às figuras. Ela é composta por três classes auxiliares: Translate, Rotate e Scale, que armazenam os parâmetros necessários para deslocamento, rotação e redimensionamento de um objeto, essas transformações permitem posicionar e manipular as figuras dentro da cena, facilitando a criação de composições complexas e ajustando a aparência dos modelos.

Foi também criada a classe Group que organiza múltiplas figuras e as suas respetivas transformações dentro de uma hierarquia. Um Group pode conter várias Figures e também outros Groups, permitindo a criação de estruturas aninhadas de objetos, além disso, esta classe aplica transformações globais a todos os seus elementos, garantindo que os modelos mantenham uma relação espacial coerente. Isso significa que, ao aplicar uma transformação em um grupo, todas as figuras e subgrupos dentro dele serão afetados de maneira proporcional.

4. Alterações gerador

Aqui adicionamos o conceito Ring que é representado por uma Figure criada dinamicamente pela função generateRing. Essa função, foi adicionada ao gerador de modelos 3D com o objetivo de criar um anel tridimensional. Esta função recebe como parâmetros o raio interno, o raio externo e o número de divisões circulares, determinando a resolução do modelo geométrico gerado. A abordagem utilizada consiste em dividir a circunferência do anel em segmentos iguais, onde cada fatia é representada por dois triângulos que, juntos, formam uma faixa entre o raio interno e externo.

Para construir o anel, a função inicia definindo um ângulo inicial a e um incremento angular delta, que é obtido dividindo 2π pelo número de fatias especificado. Em seguida, um laço percorre cada fatia do anel, avançando o ângulo a a cada iteração e gerando pontos ao longo das bordas interna e externa do anel.

A estrutura do modelo geométrico gerado segue uma organização em triângulos. Para cada fatia do anel, são criados dois conjuntos de triângulos que conectam os pontos entre os raios interno e externo, garantindo uma transição suave entre cada segmento circular. Dessa forma, a superfície do anel é composta por múltiplos triângulos que juntos formam uma estrutura contínua e bem definida.

Esta função tem diversas aplicações na modelagem de cenas tridimensionais, especialmente em representações astronômicas, como os anéis de Saturno ou órbitas planetárias.

5. Conclusão

Com a implementação desta fase do trabalho, foi possível evoluir significativamente tanto na estruturação do projeto quanto na compreensão dos conceitos envolvidos na modelagem do sistema solar. A introdução do Ring como representação dos anéis planetários trouxe uma maior complexidade ao modelo, exigindo um aprofundamento na manipulação de transformações geométricas, como rotações, translações e escalas. Além disso, a reorganização e otimização do código contribuíram para um sistema mais modular e eficiente.

O aprimoramento do motor gráfico, com foco na interpretação e aplicação de transformações hierárquicas definidas em XML, permitiu consolidar conhecimentos essenciais na integração entre C++ e a manipulação de arquivos estruturados. Dessa forma, esta fase não só trouxe melhorias visuais e estruturais ao projeto, como também reforçou a compreensão sobre conceitos fundamentais de computação gráfica e programação orientada a objetos.