

**Computação Gráfica**  
Licenciatura em Ciências da Computação  
Universidade do Minho  
Fase 3 - Relatório de Desenvolimento

Hugo Costa  
(a96059)

João Gonçalo Macedo  
(a87946)

Sara Fontes  
(a92999)

4 de junho de 2023

# Capítulo 1

## Fase 3

### 1.1 Enunciado

Nesta terceira etapa foi proposto a alteração do *engine* de forma a que seja possível ler ficheiros com Bezier patches, pontos de controlo e retornar a lista de triângulos que permitem desenhar a superfície.

Em relação ao mecanismo, queremos estender os elementos de translação e rotação. Considerando a translação, um conjunto de pontos será fornecido para definir uma curva cúbica de Catmull-Rom, bem como o número de segundos para percorrer toda a curva. O objetivo é realizar animações baseadas nessas curvas. Essa transformação inclui também um campo “align” para especificar se o objeto deve ser alinhado com a curva. Os modelos podem ter uma transformação estática ou dependente do tempo, como nas fases anteriores. No nó de rotação, o ângulo pode ser substituído pelo tempo, ou seja, o número de segundos para executar uma rotação completa de 360 graus em torno do eixo especificado.

Para medir o tempo pode ser usada a função `'glutGet(GLUT_ELAPSED_TIME)'`.

**Nota:** Devido à definição das curvas de Catmull-Rom é sempre necessário um ponto inicial antes do ponto inicial da curva e outro ponto após o último segmento. O número mínimo é de 4 pontos.

Nesta fase é também necessário que os modelos sejam desenhados com VBOs, ao contrário do que se realizou nas fases anteriores.

A cena é um sistema solar dinâmico, incluindo um cometa com uma trajetória definida usando uma curva de Catmull-Rom. O cometa deve ser construído usando patches de Bezier, por exemplo, com os pontos de controle fornecidos para o teapot.

## 1.2 Apresentação das Soluções

### 1.2.1 Engine

Para melhor organização do código separamos o nosso *engine* em várias partes, nomeadamente:

- *main.cpp* onde corremos o *engine*;
- *structs.h* de onde importamos as diferentes structs, por exemplo as transformações e os modelos;
- *structs.cpp* onde definimos as structs e as funções para cálculo de matrizes e normais para as curvas e funções necessárias para as curvas CatmullRom.

Nas structs que já já estavam definidas para as transformações pelo que apenas acrescentamos os novos tipos de transformações, nomeadamente *TranslacaoTempo* e *RotacaoTempo*, que atuavam conforme um parâmetro tempo.

Para além disto, alteramos a função *getModel* no engine para devolver um vetor com pontos e criamos buffers para os vértices e um *verticesCount* para guardar os pontos e o número de pontos de cada modelo para depois preencher os mesmos.

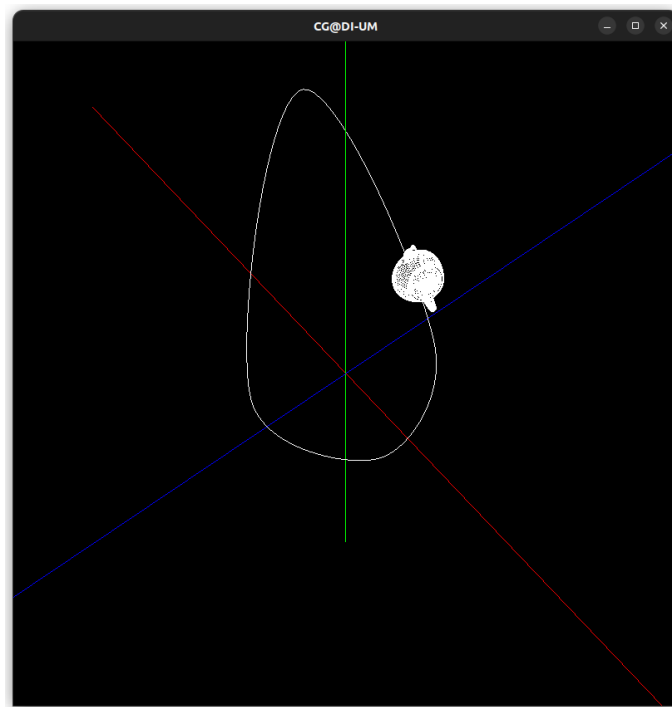
Passamos a usar vbos para guardar os vertices das figuras. Na implementação dos VBOs optou-se por utilizar um VBO para cada figura, pois pareceu ser a solução mais simples. No entanto, cada modelo diferente possui um só VBO. Por exemplo, se existirem várias esferas iguais, apenas é utilizado um VBO para todas.

### 1.2.2 Generator

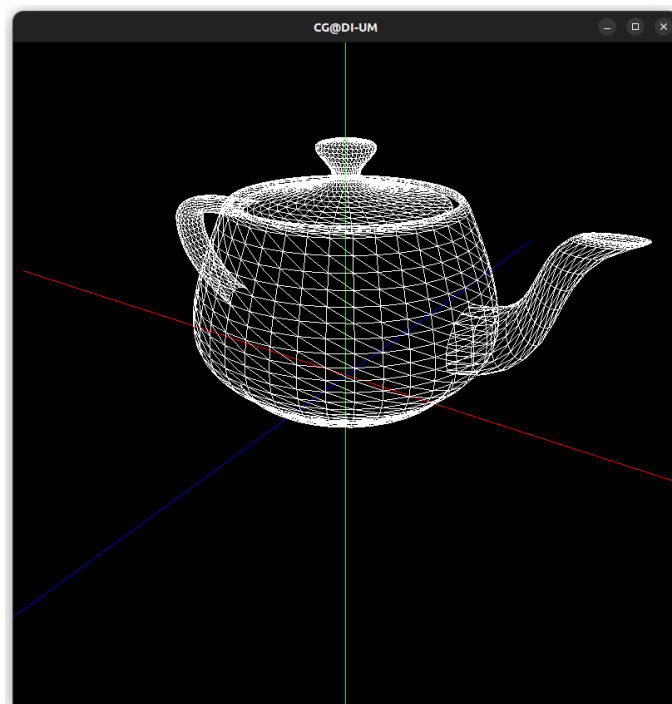
Para ser possível gerar as curvas de Bezier, começamos por criar a função *createBezier* que recebe um nome do arquivo de input e um valor de tessellation como parâmetros. Em seguida lê informações de um arquivo de patch, "teapot.patch", que contém dados sobre patches e pontos para criar uma superfície Bezier e armazenando os índices dos pontos e as coordenadas dos pontos em estruturas de dados. Em seguida, a função itera sobre os índices de patch e realiza operações de multiplicação de matriz com a função *multMatrixMatrix* para transformar as coordenadas dos pontos. Por fim, chama a função *createSurface* para gerar a superfície Bezier a partir das matrizes de coordenadas transformadas.

A função *createSurface* calcula os pontos de uma superfície Bezier com base nas matrizes de coordenadas fornecidas e retorna uma string que representa essa superfície.

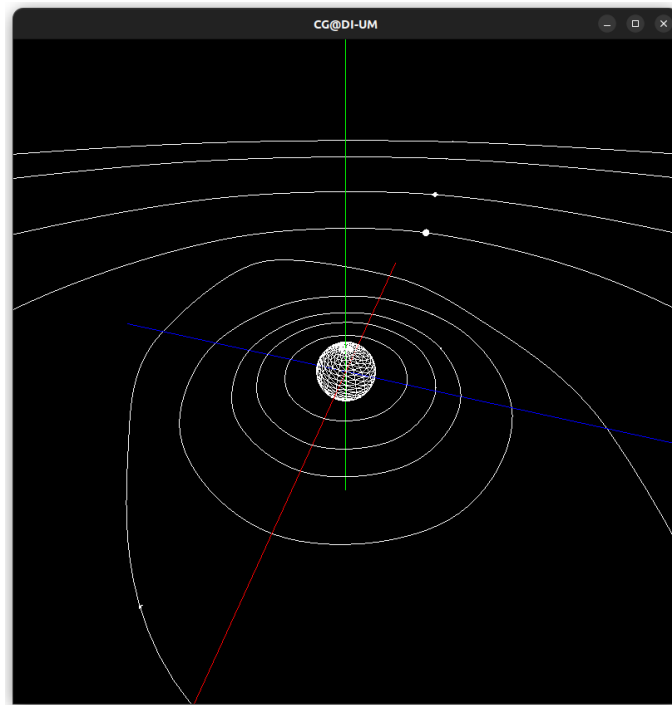
### 1.2.3 Testes e Resultados Obtidos



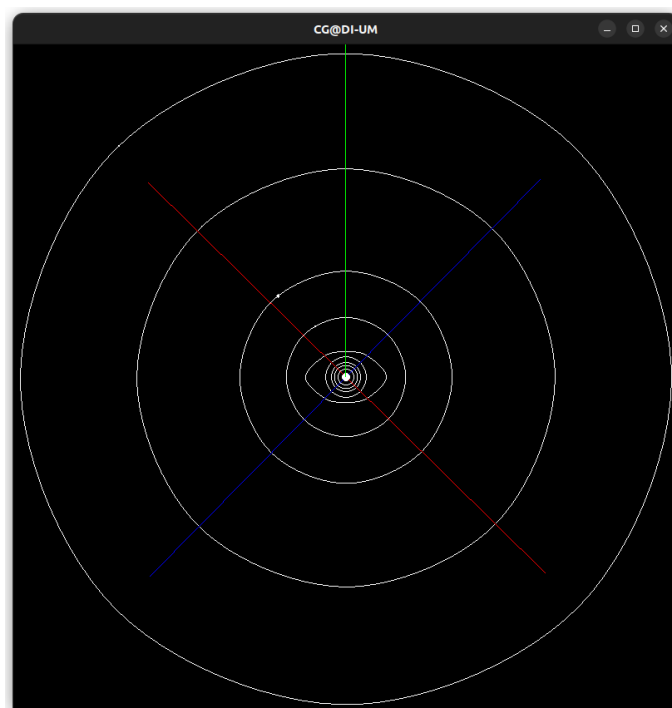
**Figura 1:** test\_3.1.xml



**Figura 2:** test\_3.2.xml



**Figura 3:** Sistema Solar com cometa



**Figura 4:** Órbitas do Sistema Solar

## Capítulo 2

# Conclusão

Durante o desenvolvimento desta fase surgiram algumas dificuldades como a implementação dos VBO's. Porém, acreditamos que completamos esta fase de forma positiva, concluindo tudo o que foi proposto.

No que diz respeito ao desenvolvimento do projeto, apesar de não termos completado a fase 4, acreditamos que conseguimos superar alguns desafios que foram surgindo ao longo de todo o processo, melhorando as nossas habilidades no que diz respeito a computação gráfica e, que entregamos um trabalho satisfatório, em que os principais objetivos foram atingidos.