



UNIVERSIDADE DO MINHO

LICENCIATURA EM ENGENHARIA INFORMÁTICA

Computação Gráfica

Trabalho Prático - Fase 3

José Freitas (A96140) Paula Marques (A90088)
Inês de Castro (A95458) Hugo Pereira (A93752)

26 de abril de 2024

Índice

1	Introdução	3
2	Trabalho Desenvolvido	4
2.1	Generator	4
2.2	Engine	5
2.2.1	VBO	5
2.3	Sistema Solar	5
2.3.1	Cometa	5
3	Conclusão	9

1 Introdução

Este trabalho, desenvolvido como parte da disciplina de Computação Gráfica, compreende duas partes principais Generator e a Engine.

Nesta fase do projeto, temos como objetivo modificar o sistema solar desenvolvido na fase anterior, para ser capaz de suportar translações e rotações.

Portanto, no que faz referencia ao Generator, para além do que fazia nas fases anteriores, consegue gerar novos modelos .3d, nomeadamente superficies de Bezier.

Quanto ao Engine, implementou-se funcionalidades capazes de criar animações como rotações e translações, com especial atenção em utilizar VBOs.

Este relatório serve de complemento ao código criado pelo grupo.

2 Trabalho Desenvolvido

À semelhança da primeira fase, esta fase, dividimos o trabalho em duas partes: Generator e Engine.

2.1 Generator

Primeiramente, como argumentado no relatório anterior, o grupo implementou um modelo novo *torus* usado para os anéis de Saturno. Para tal, recebemos como input o raio interno da circunferência, a distância entre o centro do torus e o centro da circunferência interior, o número de slices e o número de stacks. Foram implementados dois ciclos *for*, onde o primeiro percorre as slices e o segundo as stacks. São desenhados dois triângulos a cada iteração.

De modo a ser possível implementar as superfícies de Bezier para denhar a lista de triângulos, função em questão recebe o nome do ficheiro, onde se encontram os pontos de controlo de Bezier e o nível de tesselação, recebemos como input o *teapot.patch* e o nível de tesselação.

A função *bezier*, foi definida de forma a ler cada linha do ficheiro *teapot.patch*, e armazenar 16 índices de cada patch numa variável *patches* e 3 coordenadas numa variável chamada *cpoints*.

Foi criada a função *getBezierPoint* de modo a calcular os pontos de Bezier. Por sua vez, esta função recebe variáveis *u* e *v*, um float *points[16][3]* com os pontos correspondentes a cada índice de uma patch e uma *pos[3]* para armazenar o ponto calculado. Para esta função, utilizamos a seguinte expressão:

$$p(u,v) = [u^3 \quad u^2 \quad u \quad 1]M \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} M^T \begin{bmatrix} v^3 \\ v^2 \\ v \\ 1 \end{bmatrix}$$

Figura 1

Desta forma cada linha da matriz representa uma curva de Bezier composta por 4 coordenadas.

Por sua vez a Matriz *M* e a sua transposta são dadas pela seguinte expressão:

$$M = M^T \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Figura 2

2.2 Engine

Na terceira fase do Engine, esta sofreu alterações de modo a possibilitar a animação do sistema solar a partir de translações e rotações.

No que diz respeito ao parse dos ficheiros XML, de modo a implementar as novas funcionalidades foi necessário alterar o modo de leitura dos ficheiros XML, de forma a suportar transformações com tempo e pontos de controle.

Foram adicionados os vetores x, y e z para armazenar as coordenadas dos pontos de controle. Acrescentamos também a variável align para ver se o objeto está alinhado.

2.2.1 VBO

Como o uso de VBOs foi implementado nas fases anteriores, não foi necessária nenhuma alteração nesta vertente. O grupo apenas realizou esta fase como sempre tinha feito, usando VBOs, uma vez que aumenta a performance do nosso programa.

2.3 Sistema Solar

De modo a cumprir o objetivo desta terceira fase, adicionamos ao ficheiro XML as transformações necessárias para animar o Sistema Solar. O grupo teve o cuidado de utilizar tempos reais da translação e rotação dos planetas e satélites, normalizando os seus valores.

De seguida apresenta-se o Sistema solar do programa nesta fase:

2.3.1 Cometa

Ainda nesta fase foi proposta a criação de um "cometa" usando *Bezier patches*, com os pontos de controlo do *teapot*.

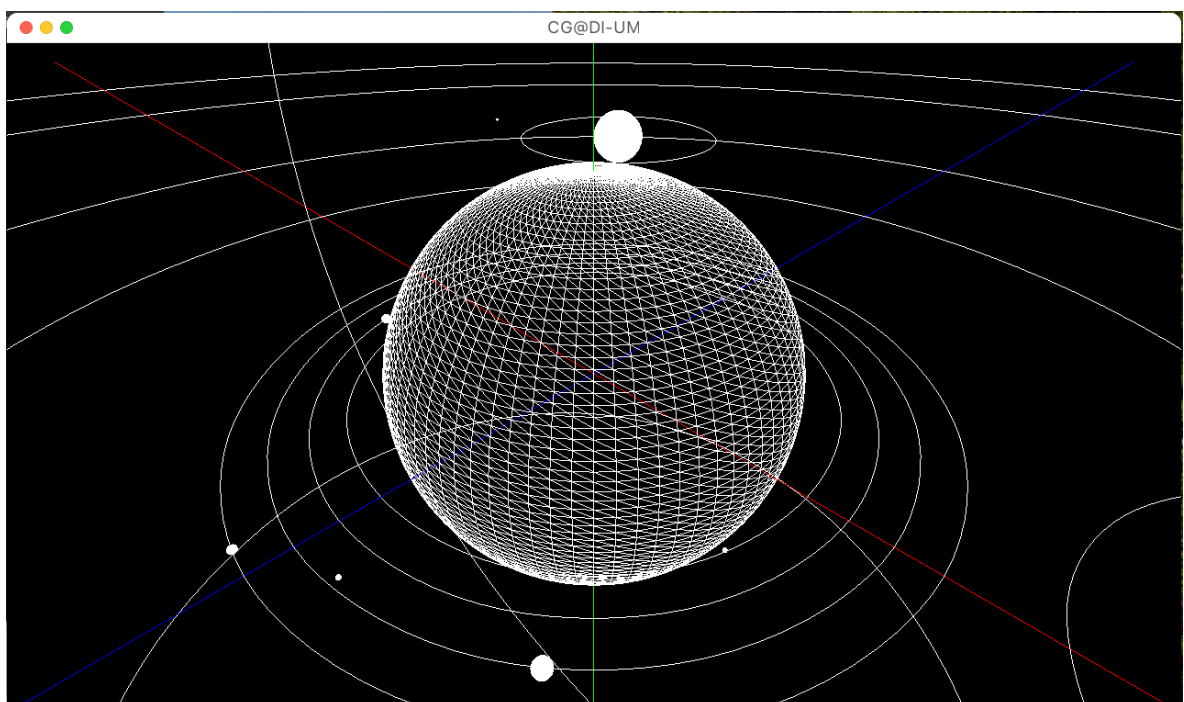


Figura 3

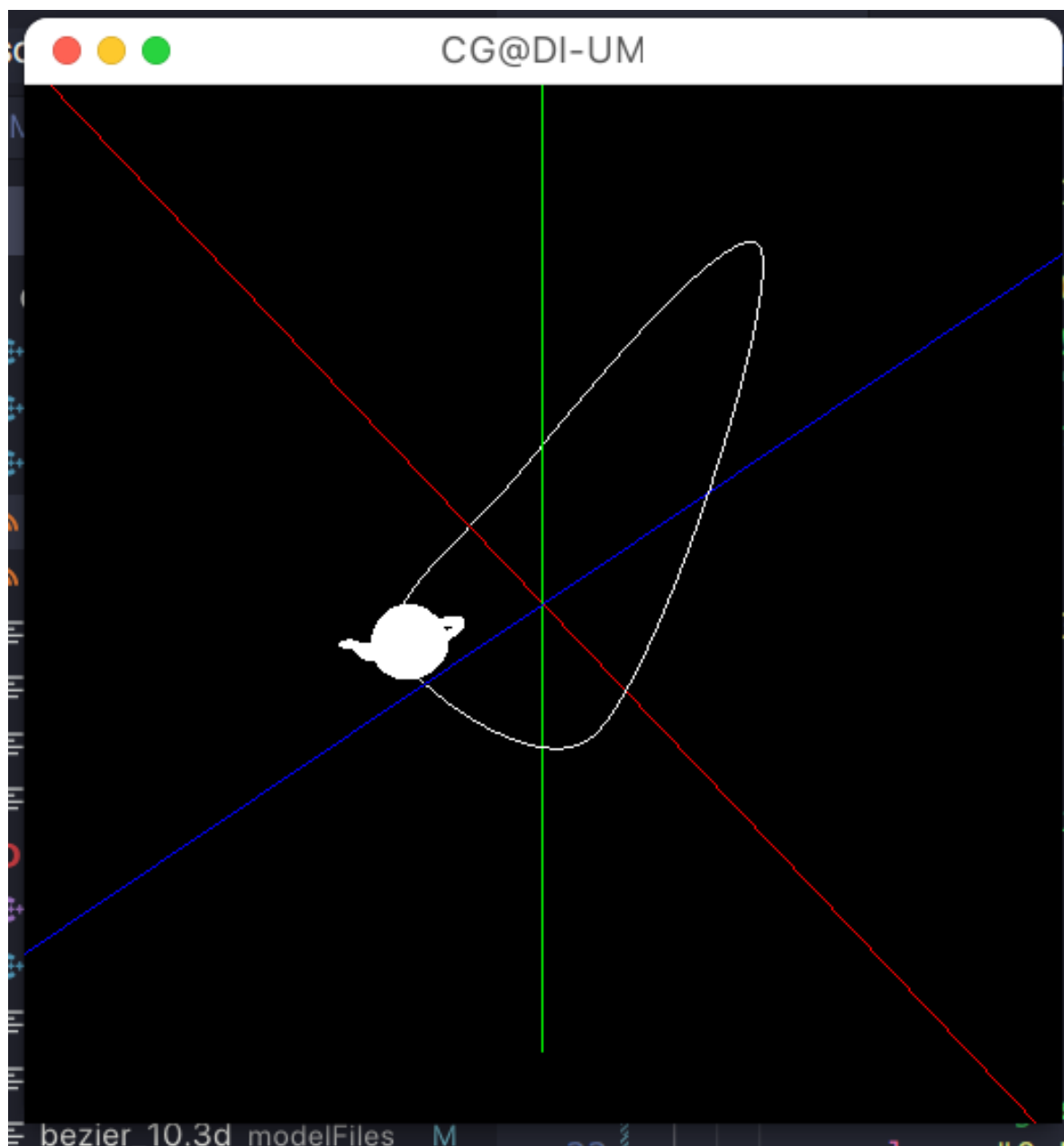


Figura 4: Resultado do teste 3.1

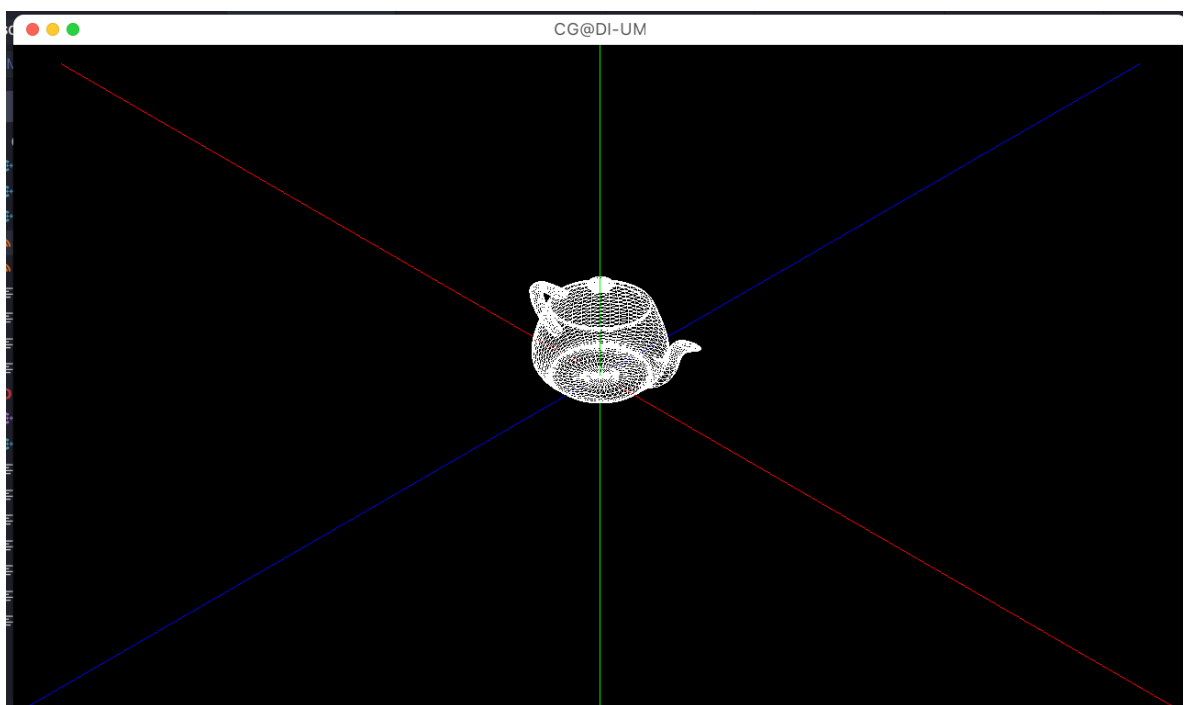


Figura 5: Resultado do teste 3_2

3 Conclusão

Este trabalho prático continua a permitir-nos aplicar os conhecimentos adquiridos durante as aulas da disciplina de Computação Gráfica, possibilitando o aprofundamento do nosso conhecimento sobre o OpenGL e o C++.

Apesar desta fase ser mais trabalhosa que a anterior e apesar das dificuldades, o grupo considera que esta fase foi concluída com sucesso, uma vez que fomos capazes de cumprir os requisitos, resultando num programa capaz de aplicar as transformações pedidas, das quais as curvas de Bezier, rotações e translações dos modelos no Sistema Solar e usar funcionalidades como o VBO.

Pretendemos melhorar o trabalho realizado nesta fase, antes da entrega da última.

Estamos confiantes que o sucesso desta fase, ajudar-nos-á na execução das próxima e última fase.