

UNIVERSIDADE DO MINHO

ESCOLA DE ENGENHARIA



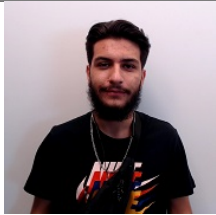


Universidade do Minho
Escola de Engenharia

Computação Gráfica

Licenciatura em Engenharia Informática

Fase 3 - Curves, Cubic Surfaces and VBOs

Grupo 39

A93221	A93256	A93169
		
André Vaz	João Mendes	Laura Rodrigues

Maio 2022

Índice

1	Introdução	3
2	Utils	3
2.1	Matrix	3
3	Generator	3
3.1	Tesselação de uma <i>Bezier Surface</i>	3
4	Engine	4
4.1	Transforms	4
4.2	VBOs	4
5	Sistema Solar	4
6	Testes	5
7	Conclusão	6

Lista de Figuras

1	Sistema Solar	5
2	test_3_0	5
3	test_3_1	6
4	test_3_2	6

1 Introdução

Esta terceira fase do projeto teve como objetivo complementar o motor gráfico desenvolvido nas fases anteriores e criar as primitivas gráficas para obter uma representação dinâmica do sistema solar. De modo a conseguir concretizar esse objetivo, foi necessário implementar curvas, superfícies cúbicas e VBOs.

2 Utils

Tal como nas fases anteriores a nossa biblioteca de funções e classes foi aumentando havendo alterações nas seguintes áreas.

2.1 Matrix

Contém funções com a intenção de realizar diferentes cálculos entre matrizes, sendo estas:

- **buildRotMatrix**: constrói a matriz de rotação que permite desenhar as várias curvas
- **toVector**: constrói um vetor a partir de dois pontos
- **cross**: cross de dois vetores
- **normalize**: normalizar os pontos
- **multMatrixVector**: este método multiplica uma matriz por um vetor

3 Generator

Visto que nas fases anteriores já tinha sido realizada a leitura de uma *bezier surface* a partir de um ficheiro, foi apenas necessário implementar a sua tesselação.

3.1 Tesselação de uma *Bezier Surface*

A partir da tesselação é criado um modelo onde as suas *patches* contêm um *array* de pontos que formam os triângulos da *patch* já tesselada.

De forma a tesselar uma *bezier patch* (que contém 16 pontos), foram usados 4 pontos de controlo de cada vez. Para cada um foram calculadas as quatro primeiras curvas de bezier (ou seja para cada nível de tesselação) que por sua vez permitiram calcular a secção da superfície. Tendo estes quatro pontos, pudemos organizá-los de forma a que estes formassem os dois triângulos de cada secção. A leitura e escrita do modelo tesselado é realizado da mesma forma que em fases anteriores, através da *readModel* e da *writeToFile*.

4 Engine

Na continuidade das fases anteriores do projeto, foi adicionado uma funcionalidade de movimento dos modelos usando a curva de *Catmull-Rom* e a renderização de modelos por *VBOs*

4.1 Transforms

Relativamente à fase anterior, acrescentámos novas variáveis como o *time* : *float*, *align* : *boolean* e o *path* : *vector*< *Point* > à classe *Translate* com o intuito de realizar animações.

Assim sendo, no ficheiro XML o *Translate* pode possuir um atributo *time* que indica a duração da animação e possui em si elementos *child* sendo estes os diferentes pontos da trajetória.

Para além disso, tivemos de elaborar funções responsáveis pela elaboração da curva *Catmull-Rom*, onde conseguimos calcular a posição do grupo a ser desenhado em determinado instante.

Para calcular os pontos para a implementação das curvas baseamo-nos na equação que se segue

$$\begin{bmatrix} t^3 & t^2 & -t & 1 \end{bmatrix} \cdot \begin{bmatrix} -1/2 & 3/2 & -3/2 & 1/2 \\ 1 & -5/2 & 2 & -1/2 \\ -1/2 & 0 & 1/2 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} x(u) & y(u) & z(u) \end{bmatrix}$$

4.2 VBOs

De forma a renderizar os modelos, os seus vértices são adicionados a um *array* de *floats*, de seguida é criado um *buffer* responsável por guardar os vértices de determinado modelo, sendo o mesmo populado com o *array* previamente criado que por sua vez é libertado de memória. Aquando de desenhar o modelo, a esse *buffer* é dado *bind*, garantindo que é de facto um *array* de vértices com 3 dimensões e de seguida desenhado usando a função *glDrawArrays*.

5 Sistema Solar

Na elaboração do sistema Solar relativamente à fase anterior implementámos um cometa (*Teapot*) com uma trajetória usando uma curva de *Catmull-Rom* (já referida anteriormente). Como podemos ver na imagem que se segue, esta tem 3 pontos.

De forma a complementar, adicionámos os anéis de Saturno. De forma a permitir que estes sejam visíveis de cima e baixo foi necessário introduzir dois anéis (um virado para cima e outro para baixo).

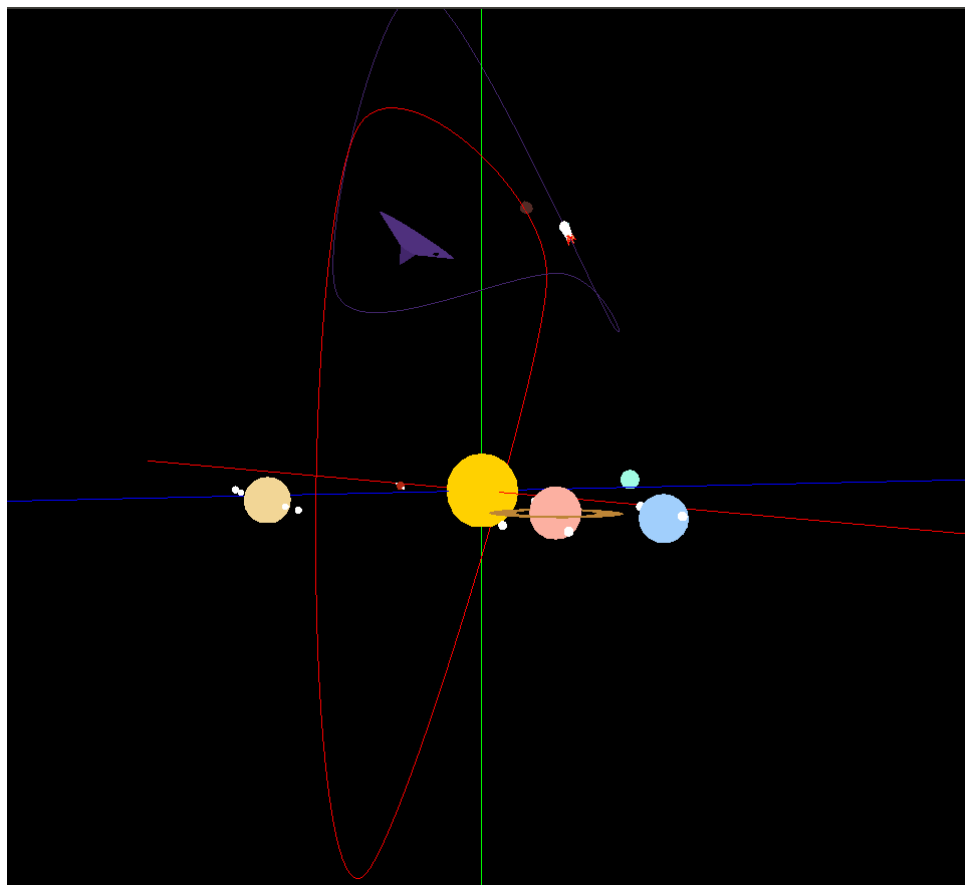


Figura 1: Sistema Solar

Adicionalmente criámos uma trajetória para o foguetão criado como extra na fase anterior. Por sua vez, esta curva de *Catmull-Rom* é feita a partir de 4 pontos. Analisando o ficheiro SistemaSolar.xml podemos observar que todos os planetas têm rotações em volta do sol para o mesmo sentido, excetuando Vénus e Urano, que à semelhança do que acontece na realidade têm uma rotação retrógrada (no sentido contrário).

6 Testes

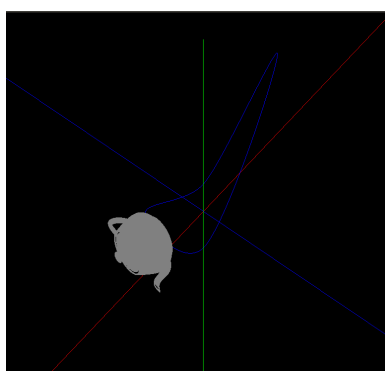


Figura 2: test_3_0

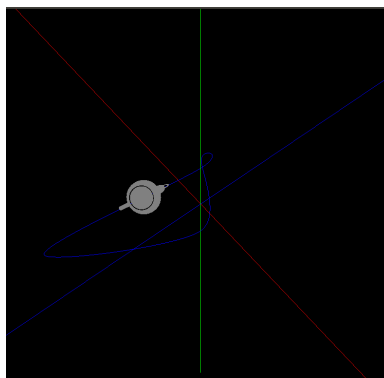


Figura 3: test_3_1

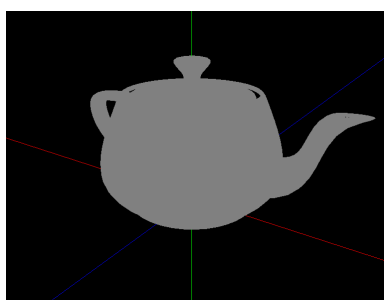


Figura 4: test_3_2

7 Conclusão

O projeto teve como intuito melhorar o motor gráfico desenvolvido nas fases anteriores e obter uma representação dinâmica e realista do Sistema Solar.

Nesta fase do projeto, sentimmos algumas dificuldades relativamente à implementação da *Tessellation*, o que causou um atraso na entrega do projeto.

Posto isto, houve o cuidado de cumprir todos os parâmetros pedidos bem como elaborar requisitos extras, como a implementação dos anéis de Saturno bem como a *MouseMotion*.

Por fim, consideramos que houve um balanço positivo, uma vez que ultrapassámos com sucesso as dificuldades sentidas na elaboração, acabando por cumprir todos os requisitos pedidos, apesar do projeto ter sido entregue com algum atraso.