Universidade do Minho

Escola de Engenharia

Computação Gráfica

Licenciatura em Engenharia Informática

Fase 3 – Curves, Cubic Surfaces and VBOs

Grupo 47

Alexandre Fernandes - [A94154]
Henrique Vaz - [A95533]
Vasco Rito - [A98728]
Pedro Oliveira [A98712]

Maio, 2023

Conteúdo

1. Introdução

Nesta terceira fase do projeto estendemos as capacidades do motor gráfico adcionando animações e novas figuras (como por exemplo o meteorito que foi desenhado com *patches*).

O *generator* foi também estendido para processar *Bezier Patches* e gerar os ficheiros *3d* equivalentes de forma a ser mais simples adicionar modelos complexos sem uso das primitivas.

Por fim, o modelo de teste utilizado (O Sistema Solar) foi alterado para demonstrar estas capacidades, fazendo com que os planetas orbitassem o Sol e com que as luas orbitassem os planetas, contando ainda com a adição de um cometa.

2. Generator

Nesta fase, foi-nos proposto a alteração do *Generator* de modo a que este fosse capaz de criar novos tipos de modelo utilizando, para isso, *patches* de *Bezier*.

Com esta nova estratégia, somos agora capazes de gerar primitivas com um grau de complexidade bem mais elevado às previamente disponíveis nas anteriores fases do projeto, sendo-nos também possível especificar o grau de detalhe com que desejamos gerar estas novas primitivas.

2.1 Patch

Um patch é um conjunto de 16 pontos que definem uma superfície de *Bezier*, sendo que , na verdade, cada *patch* corresponde a 16 índices que referenciam pontos de controlo da superfície.

Cada *patch* destes representa uma superfície e os 16 pontos são divididos em grupos de 4 que descrevem arcos desta superfície. Depois, para cada arco definem-se os pontos da curva usando o nível de *tessellation* para definir quantos pontos criar. Por fim, utilizando estes pontos aplica-se o mesmo algoritmo para definir N curvas (uma para cada 4 pontos de cada arco) unindo os 4 arcos e formando uma rede de arcos. Os quadrados desta rede são mais tarde partidos em dois triângulos e serializados como 6 pontos no espaço.

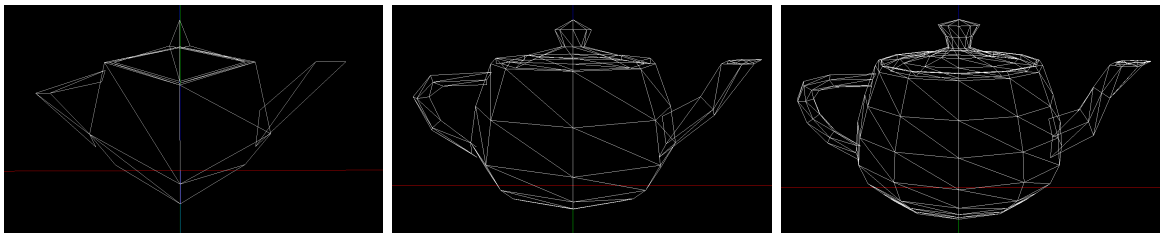

Figura 2.1: Teapot gerado com 3 níveis de detalhe distintos (tesselation 1,2 e 3, respetivamente)

3. Arquitetura do Projeto

3.1 Model

Nesta fase os modelos passaram a ser desenhados com *Vertex Buffer Objects* (VBOs). Com a sua utilização, conseguiu-se aumentar, significativamente, a *performance* do programa. Esta melhoria é conseguida pois é diminuído o números de pedidos à placa gráfica.

3.2 Transformations

De forma a animar os modelos, foram criadas novas versões das transformações, a **TranslateDynamic** e a **RotateDynamic**. Estas funções recebem o tempo que demoram a completar a animação e, depois de terminada, recomeça do início.

3.2.1 TranslateDynamic

A função **TranslateDynamic** recebe um conjunto de pontos que definem o caminho por onde o modelo vai passar. Depois, usando o número de milissegundos desde a execução do programa e fazendo uso do algoritmo de *Catmul-Rom*, é calculado o ponto seguinte para onde, o modelo, se deve mover.

3.2.2 RotateDynamic

A função de rotação dinâmica tem os parâmetros muito similares à sua versão estática, com a ligeira diferença de que, em vez do ângulo de rotação, recebe a duração da animação. Esta duração representa o tempo que o objeto demora a efetuar uma rotação de $360^{\underline{o}}$.

O ângulo de rotação é calculado da seguinte forma:

$$angle = \frac{elapsed * 360}{duracao} \tag{3.1}$$

3.2.3 Group

O método para desenhar recebe, agora, o tempo que passou desde o início do programa, para que o possa passar às várias transformações. Desta forma, é assegurado que todas as transformações usam o mesmo valor de tempo.

4. Ficheiro XML

O XML de *input* utilizado foi estendido para possibilitar a utilização das novas funcionalidades, no entanto, foi mantida a compatibilidade para os formatos anteriores. De forma a que modelos antigos continuem a ser validos.

4.1 Translação Dinâmica

Para definir uma translação dinâmica cria-se um bloco *translate* com o parâmetro *time*, e dentro deste define se a lista de pontos que a translação irá seguir.

```
<translate time="30.99180855951996">
    <point x="1.8" y="0" z="0.0"/>
    <point x="1.6443818237566816" y="0" z="0.7321259575364403"/>
    <point x="1.204435091445945" y="0" z="1.3376606858593094"/>
    <point x="0.5562305898749055" y="0" z="1.7119017293312764"/>
    <point x="-0.18815123388177601" y="0" z="1.7901394116628921"/>
    <point x="-0.8999999999999996" y="0" z="1.5588457268119897"/>
    <point x="-1.4562305898749053" y="0" z="1.0580134541264519"/>
    <point x="-1.760665681320 85" y="0" z="0.3742410434719675"/>
    <point x="-1.7606656813208503" y="0" z="-0.37424104347196635"/>
    <point x="-1.4562305898749057" y="0" z="-1.0580134541264514"/>
    <point x="-0.9000000000000008" y="0" z="-1.558845726811989"/>
    <point x="-0.18815123388177762" y="0" z="-1.790139411662892"/>
    <point x="0.556230589874905" y="0" z="-1.7119017293312766"/>
    <point x="1.204435091445944" y="0" z="-1.3376606858593103"/>
    <point x="1.644381823756681" y="0" z="-0.7321259575364417"/>
</translate>
```

Figura 4.1: Exemplo de uma translação dinâmica

As translações dinâmicas têm também a particularidade de possuirem um campo *align*, que especifica se os objetos de um dado grupo devem estar alinhados com a curva em questão.

```
<translate time="30.99180855951996" align = "true">
```

Figura 4.2: Exemplo de uma translação dinâmica com align

Visto estarmos a trabalhar com curvas de *Catmull-Rom*, e que em cada instante realizamos o cálculo da nova posição da curva, assim como a sua derivada, apenas precisamos de realizar uma simples rotação de -90 graus em relação ao vetor tangente (derivada) para obtermos os resultados da anterior funcionalidade.

4.2 Rotação Dinâmica

Uma rotação dinâmica apenas têm o tempo de duração de uma "volta".

```
<rotate time="10" x="0" y="1" z="0" />
```

Figura 4.3: Exemplo de uma rotação dinâmica

5. Sistema Solar

5.1 Alterações ao Sistema Solar

De forma a obter um modelo do Sistema Solar que seja o mais próximo da realidade, as transformações têm em conta a escala real. No entanto, algumas das distâncias e escalas foram alteradas para termos uma visão mais agradável de todo o sistema solar.

No desenho dos planetas todos têm por base a mesma esfera (ou torus quando usado) que depois sofrem as transformações necessárias para obtermos a posição e escala desejada. Nesta fase foram também adicionados asteroides que "circulam"pelo sistema solar.

Todos os planetas orbitam em volta do sol e giram em torno do seu próprio eixo. Adicionalmente alguns planetas têm luas que orbitam à sua volta.

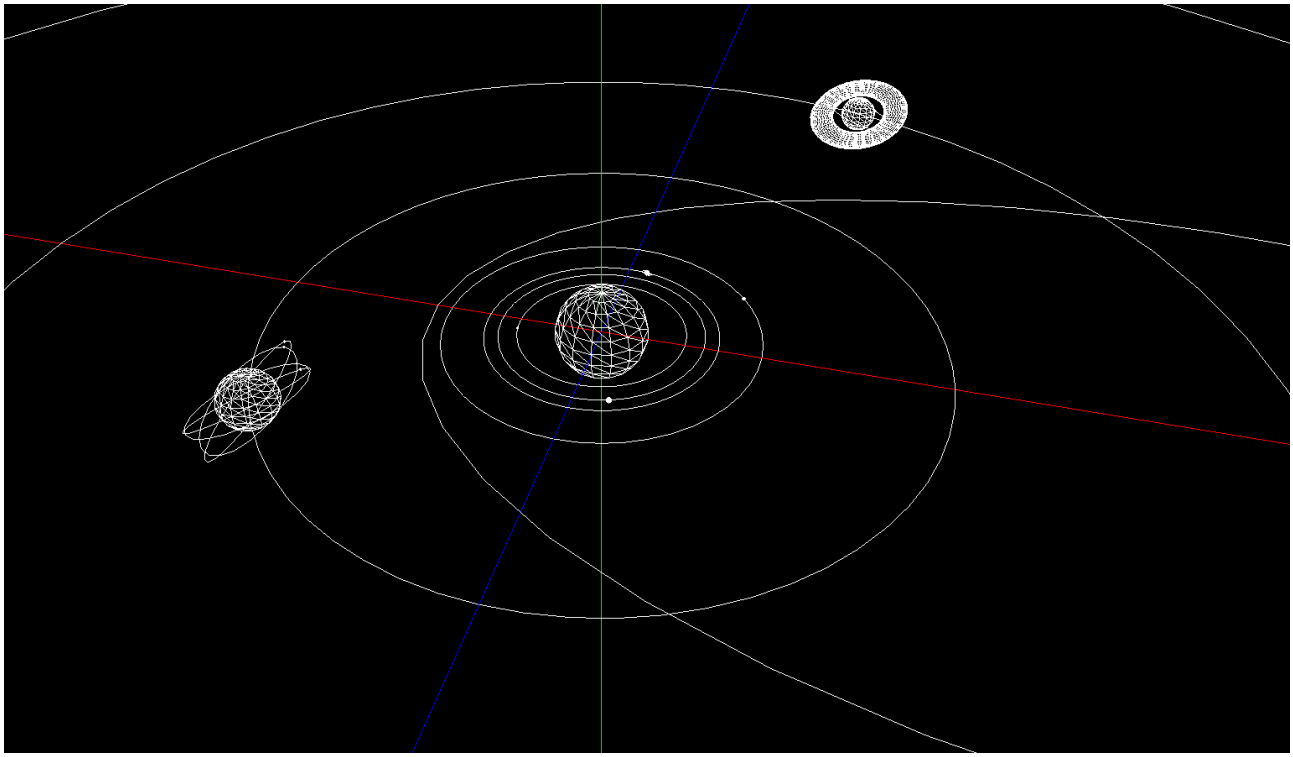

Figura 5.1: Sistema solar dinâmico com as respetivas curvas

5.2 Design do Cometa

Para o desenho do cometa (por razões de falta de tempo) o grupo decidiu usar um ficheiro patch de um elefante que estava disponível na internet, acrescentando posteriormente uma plataforma por baixo do mesmo. Como a escala do cometa é relativamente pequena o aspeto do mesmo torna-se quase impretceptivel.

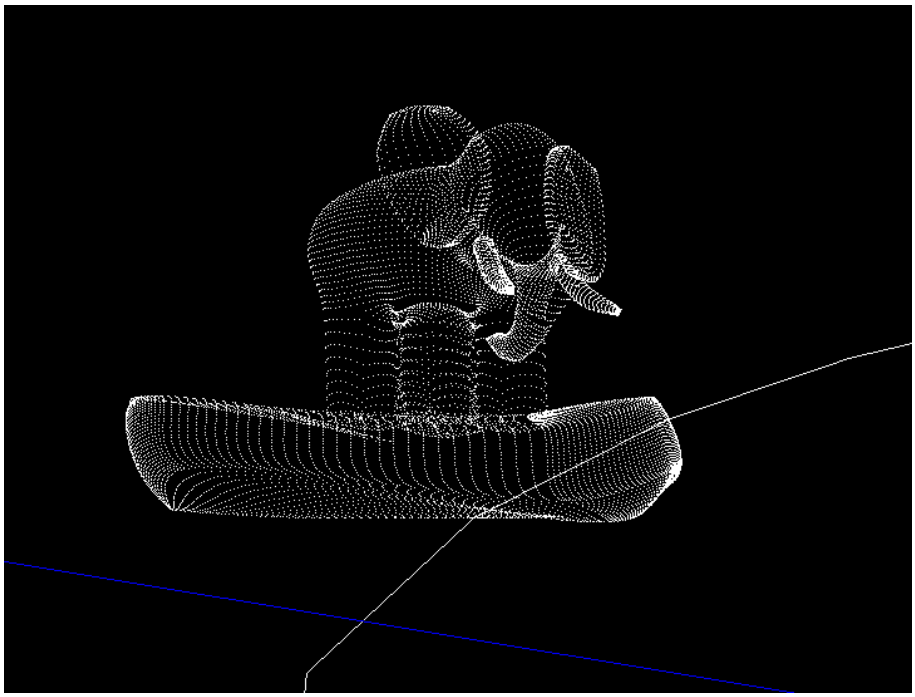

Figura 5.2: Cometa

6. Câmara

Para facilitar a visualização das figuras, foi implementada a possibilidade de mover tanto a câmara como as figuras, tanto através do uso do teclado ou rato. É também importante notar, que esta funcionalidade se destina maioritariamente ao auxílio em momentos de *debug* e exploração de cenas criadas.

As teclas associadas a cada movimento são:

- **W,S,A,D** : Alterar o ponto para o qual a câmara "olha"individualmente em cada eixo.
- **+,-** : *Zoom in* e *zoom out*.
- **F,L,P** : Modo *Fill*, *Line* e *Point*.
- **B,N,M** : GL_BACK, GL_FRONT e GL_BACK_AND_FRONT.
- **Botão esquerdo do rato + arrastar** : Rotação da câmara em relação a cada eixo.
- **Botão direito do rato + arrastar** : *Zoom in* e *zoom out*.

7. Conclusão

Em suma, todas as funcionalidades pedidas para a fase 3 foram implementadas completamente. Além do sistema solar já ter todas as orbitas a funcionar, foi também aumentada a eficiência do programa com o uso dos VBOs.

Com a realização desta terceira fase, o grupo consolidou melhor os conhecimentos tanto em relação à forma de como implementar o uso dos VBOs, bem como a eficiência acrescida do seu uso, também entedemos melhor a matéria de Curvas e Superfícies, aonde esta foi utilizada para fazer as órbitas dos planetas.