

Universidade do Minho

Unidade Curricular:

Computação Gráfica

Relatório Fase 3 - Projeto Prático

Elementos do grupo:

A93253: David Alexandre Ferreira Duarte

A89518: Ema Isabel Quintãos Dias

A94166: Samuel de Almeida Simões Lira

Índice

Índice	1
Introdução	2
Classes	3
Class Translate	3
Class Rotate	3
Class VBO	3
Raciocínio para fazer animações baseadas nas curvas cúbicas de Catmull-Rom	4
Resultados obtidos	5
Problemas e dificuldades	7
Sistema Solar	7

Introdução

Para a fase 3 do projeto foi solicitado que o *generator* criasse um novo tipo de modelo com base em *patches* de *Bezier*. O *generator* deve receber como parâmetros o nome do ficheiro onde os pontos de controle de *Bezier* são definidos, assim como o nível de tesselação. De tal forma, que o ficheiro produzido contenha os triângulos para desenhar a superfície.

Em relação à *engine* foi requisitado que se expanda os elementos *translate* e de *rotate*. Para a translação um número de pontos será fornecido para definir a curva cúbica de *Catmull-Rom*, assim como o número de segundos para percorrer a curva toda, pois o objetivo é realizar animações com base nas curvas. Esta transformação tem de incluir também um campo *align* que especifica se o objeto está alinhado com a curva. Os modelos podem ter uma transformação em relação ao tempo ou uma transformação estática. É de mencionar também que no nodo *rotation* o ângulo pode ser substituído pelo tempo, ou seja, o número de segundos que demora a fazer uma rotação de 360 graus sobre um eixo específico.

Nesta fase é necessário que os modelos sejam desenhados com **VBOs**, ao contrário da forma imediata como nas fases anteriores.

Classes

Para atingir os objetivos, foi necessário criar três novas classes a fim de facilitar respectivamente as operações de translação, rotação e desenho dos vértices através de *VBOs*. As classes *translate* e *rotate* servem como estruturas de dados que guardam os dados lidos no ficheiro xml através da função `readGroup`.

Class Translate

Esta classe tem como objetivo permitir a aplicação correta das transformações do tipo *translate*, com os campos *time* (tempo para completar o trajeto na curva), *p* (vetor com os pontos de controlo) e *align* (*boolean* que diz se o objeto tem de estar alinhado com a curva ou não).

Class Rotate

Esta classe tem como objetivo permitir a aplicação correta das transformações do tipo *rotate*, com os campos *rtime* (*boolean* que indica se o *rotate* contém um *time* ou *angle*), *angle* (ângulo em graus da rotação a ser efetuada), e os campos que representam as coordenadas do eixo de rotação, *x*, *y* e *z*.

Class VBO

Esta classe foi criada com a iniciativa de ajudar no preenchimento das várias **VBOs** ao longo da leitura dos grupos no *parsing* do ficheiro *xml*. A classe **VBO** contém o número de vértices do tipo *GLuint* assim como dois índices, o *indexB* para os irmãos e o *indexC* para os filhos .

Raciocínio para fazer animações baseadas nas curvas cúbicas de Catmull-Rom

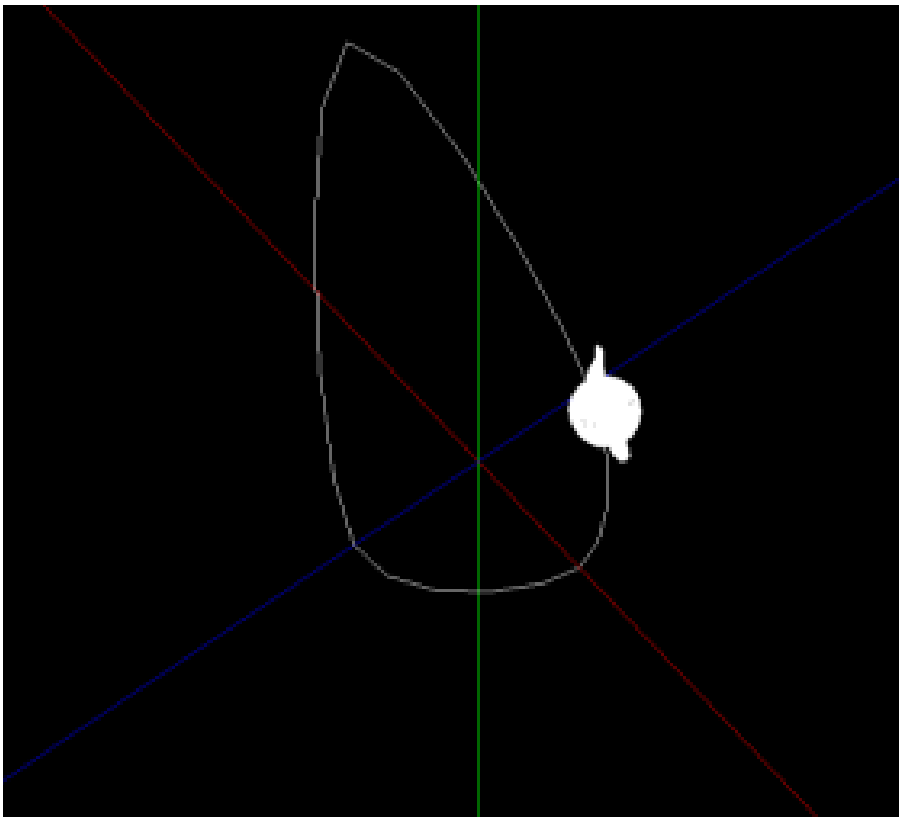
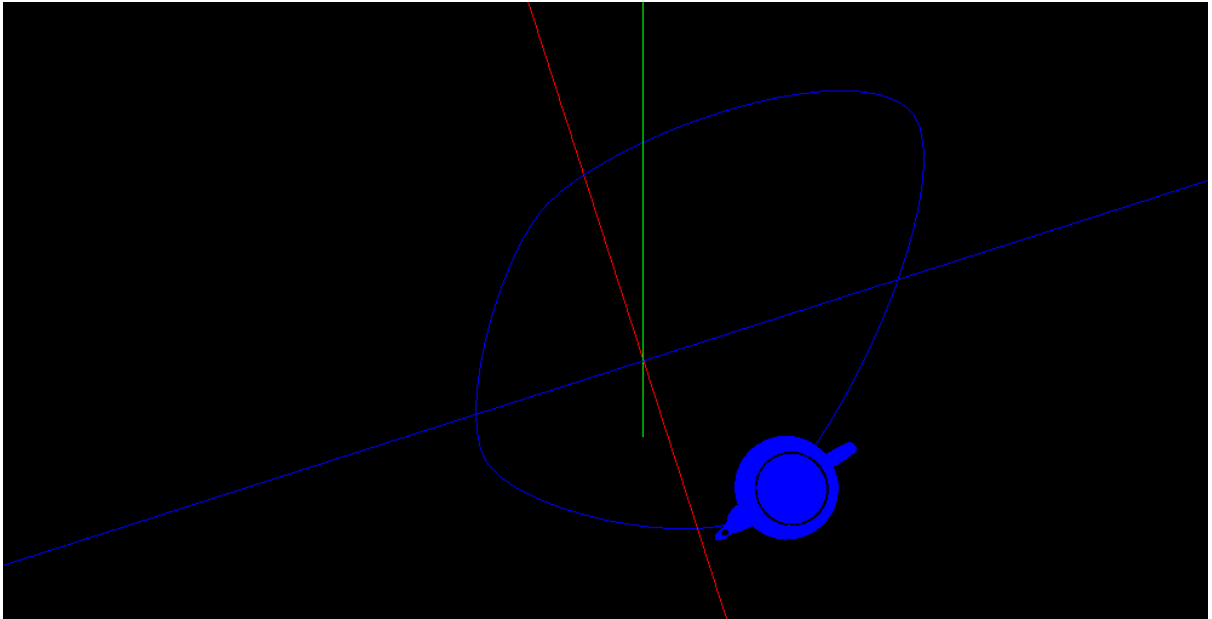
Para criar o efeito de movimento no nossos modelos, a utilização da *glutIdleFunc(renderScene)* possibilita que o *renderScene* seja chamado constantemente, ao unirmos isso com uma translação, auxiliada com a alteração da tesselação a cada *frame*, ou seja, a cada *frame* é calculada uma nova posição do objeto na curva, para ser possível realizar a animação do mesmo *frame* a *frame*. Para alcançarmos, isto a função *draw* que tem como finalidade desenhar um grupo com todas as suas transformações, chama a função *t_aux_apply* que calcula os ponto atual do objeto conforme o nível de tesselação atual, nível esse que varia de acordo com o tempo lido no *xml*.

Resultados obtidos

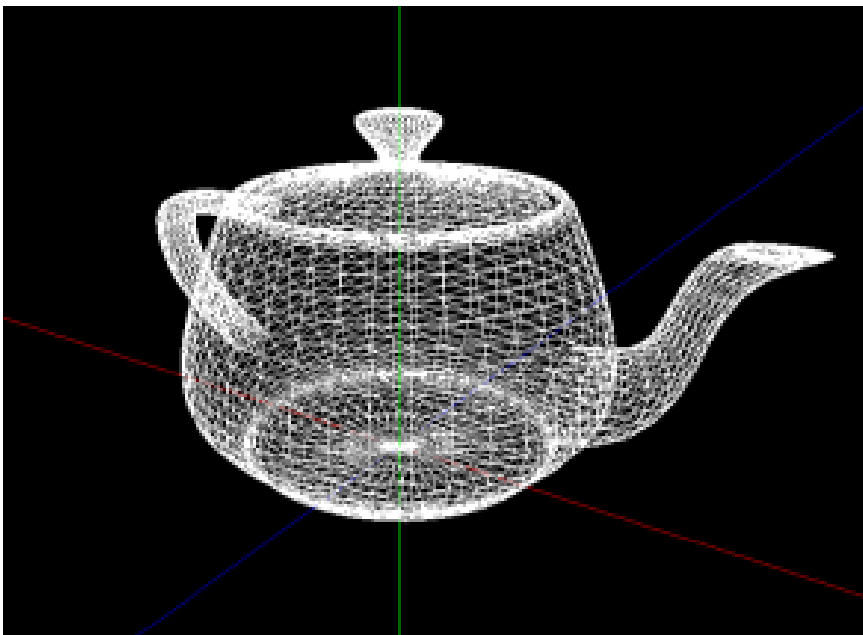
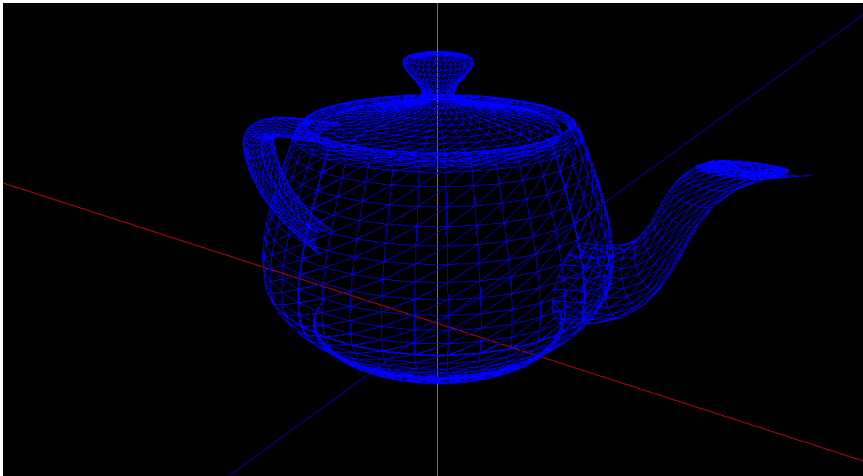
A equipa antes de criar o ficheiro *xml* que deveria representar o sistema solar, testou o projeto com os ficheiros disponibilizados pela equipa docente, obtendo os seguintes resultados:

Por cima, com o desenho na cor azul podemos encontrar os resultados da equipa face ao ficheiro teste que se pode visualizar na parte de baixo com a imagem da figura na cor branca.

test_3_1.xml



test_3_2.xml



Problemas e dificuldades

Em relação às **VBOs**, o grupo sentiu dificuldade na organização das mesmas com os respectivos *IDs*. Tal acontece pois no ficheiro *xml* o *group* pai está a utilizar os *buffers* dos filhos devido ao errado uso dos *IDs*, assim como o *filho_(n+1)* corrompe o **VBO** do *filho_n* impedindo assim a correta renderização dos *n* filhos anteriores. Apesar disso, a equipa forneceu de forma coerente os ids, onde para cada id de filho se itera conforme o do pai + 1.

Relativo ao cenário do sistema solar, o grupo teve dificuldades em alterar a escala para cada grupo distinto, problema que não ocorreu nas fases anteriores, sentiu-se também dificuldades em definir diferentes tempos entre os grupos. Estes problemas devem-se ao fato do grupo ter levado muito tempo para diagnosticar esses problemas, levando a impossibilidade deles serem resolvidos antes da entrega desta fase.

Sistema Solar

Para o sistema solar, a equipa começou por obter os pontos para as curvas de Catmull-Rom que indicariam a órbita dos planetas. Como mencionado anteriormente, nos **Problemas e dificuldades**, a equipa não conseguiu utilizar como recurso a ferramenta *scale* para diminuir ou aumentar o tamanho dos planetas, luas e do sol. Consequentemente, tal problema impossibilitou a construção do sistema solar com luas, e com os anéis de Saturno. Assim, o sistema solar construído contém os corpos celestes todos do mesmo tamanho e os planetas a percorrer a sua órbita à volta do sol, rodando sobre o próprio eixo enquanto orbitam, em tempos distintos para cada corpo celeste.

