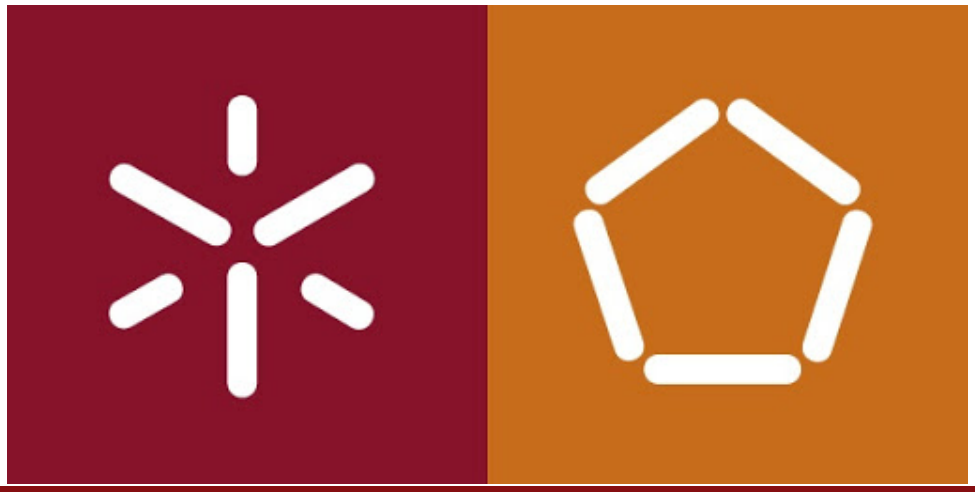

Computação Gráfica
Universidade do Minho

Henrique Paz (a84372), João Queirós (a82422), José Santos (a84288), Pedro Gomes (a84220)

23 de Julho de 2020

Conteúdo

1 Introdução

Para a quarta e última fase do trabalho, foi necessário adicionar iluminação e texturas, sendo que deste modo, ao generator foram adicionadas as funcionalidades para gerar normais de cada vértice nos ficheiros .3d, sendo que o engine passou a ser capaz de processar os novos dados assim como processar a informação acrescentada nos ficheiros xml, nomeadamente as imagens para textura e a iluminação.

2 Generator

No generator, foi necessário adicionar ás várias primitivas as operações necessárias para incluir as normais dos pontos, assim como as coordenadas de textura, sendo que não houve necessidade de alterar aquilo que tinha sido desenvolvido anteriormente.

2.1 Plano

As normais do plano, sendo que este está sempre assente no plano XoZ, são iguais para todos os pontos, e têm de valor (0,1,0). As coordenadas de textura correspondem aos 4 pontos do plano, sendo que cada um deles corresponde a uma coordenada de textura.

2.2 Caixa

As normais da caixa podem ser divididas segundo as suas faces, sendo que para cada uma delas, as normais correspondentes são:

1. Face frontal: (0,0,1).
2. Face traseira: (0,0,-1).
3. Face de cima: (0,1,0).
4. Face de baixo: (0,-1,0).
5. Face da esquerda: (-1,0,0).
6. Face da direita: (1,0,0).

Para as coordenadas de textura, a cada vértice (i,j) é atribuída a coordenada (i/divisões, j/divisões), sendo que as divisões correspondem ao número de divisões indicadas para gerar a caixa.

2.3 Esfera

Na esfera as normais de cada vértice são obtidas através da normalização do x,y,z do ponto, ou seja, divindo o valor pelo raio. Em relação às cordenadas de textura para cada vertice (x,y) a textura é ((slices -y)/slices e (stacks - x)/stacks).

2.4 Patches de Bezier

Para calcular as normais de cada ponto de uma superfície de bezier fazemos o cálculo das derivadas parciais. Tendo esse valor, o vector normal a um dado ponto da superficie é o resultado normalizado do produto externo dos vetores tangentes ao ponto. Em relaçao às texturas a um ponto corresponde a textura uma vez que os dois variam entre 0 e 1.

3 Engine

3.1 Estruturas de Dados

3.1.1 Model.cpp

Na parte do engine fizemos alterações a algumas estruturas como por exemplo a do model em que acrencentamos dois arrays de vértices que são preenchidos na leitura do ficheiro .3d e que ficam preenchidos com os pontos das normais e os pontos de textura. Também adicionamos buffers para as texturas e as normais para a criação dos VBOs das texturas e dos pontos normais. Por fim também guardamos qual a textura que o modelo tem e um id que serve como identificaçao na criaçao dos VBOs.

3.1.2 Shine.cpp

Além disso o modelo agora pode ter um Shine que consiste nas quatro componentes da iluminaçao, difusa,especular,emissiva e ambiente. Neste caso só usamos para o modelo do sol no xml visto que é um componente que tem luz própria, e como a luz do tipo POINT está dentro do sol, esta não o ilumina logo o sol precisa de outra fonte de iluminaçao.

3.1.3 Lights.cpp

Além disso fizemos um novo modelo, o lights que vai conter cada luz especificadas no modelo .xml do sistema solar com a tag <lights> que pode ser de vários tipos, POINT, DIRECTIONAL ou SPOT tendo cada uma as suas espeficiações diferentes.

4 Sistema Solar

Nesta fase, todos os elementos do sistema solar possuem texturas, assim como as normais necessárias para a iluminação destes. Para representar o sol, e visto que a luz era gerada no centro deste, foi inicialmente gerado todo o resto do sistema solar, incluindo a iluminação, depois, para o sol, foi desativada a iluminação e aí é que o sol foi gerado, sendo a iluminação reativada posteriormente, pois, caso o sol fosse gerado com os restantes elementos do sistema, este não seria iluminado pois a luz está dentro do modelo.

Figura 1: Sistema solar e cometa teapot

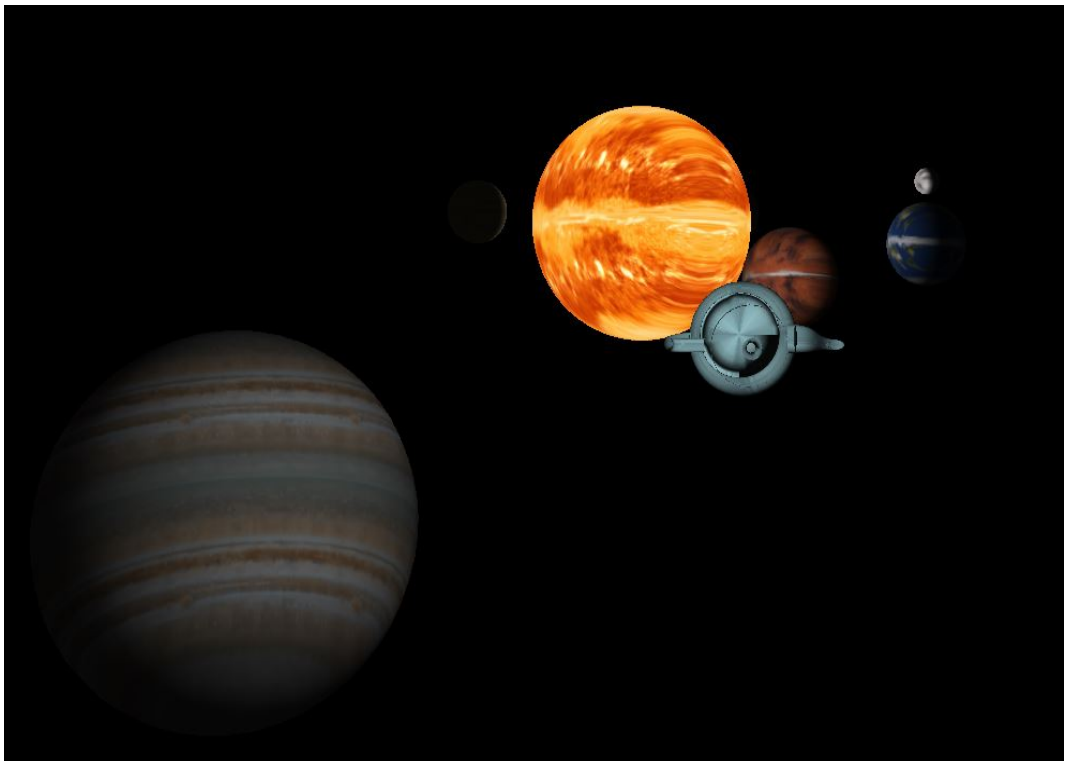

Figura 2: Sistema solar e cometa teapot

5 Conclusao

Para concluir esta 4ª fase e o trabalho de computação gráfica o nosso grupo considera que tenha tido um resultado positivo visto que o sistema solar contem todos os planetas com as animações de translação, escala, rotação com as devidas texturas e com iluminação.Tudo isto usando VBOs para uma maior eficiência na visualização do modelo. Posto isto achamos que também existe espaço para melhorar principalmente na parte de iluminação que achamos que podia estar melhor mas o tempo não nos permitiu mais. Por fim achamos este trabalho muito ilucidativo em relação à geração de modelos gráficos usando o opengl e temos a certeza que esta experiência vai ser útil num futuro próximo.