Computação Gráfica

Phase 3

GRUPO 6
A95917 - Eduardo Diogo Costa Soares - LCC
A95609 - Duarte Alexandre Oliveira Faria - LEI
A97941 - Diogo Filipe Oliveira da Silva - LEI
A98979 - Pedro Domingues Viana - LCC

Introdução

Nesta terceira fase do projeto foram modificados:

- o gerador para que este consiga ler um patch de Bézier e gerar um ficheiro .3d com os vértices que formam o modelo.
- o engine consegue ler as novas extensões dos elementos *rotate* e *translate* e foi também aplicado um novo método de desenho com uso de VBOs.

Mudanças no generator

Como já foi mencionado o generator foi alterado de forma a que este consiga ler ficheiros que contêm um patch de Bézier. Para isto é utilizada a função bezier(file_in, tessellation, file_out) responsável por ler o ficheiro .patch que contém a definição dos patches e as coordenadas dos pontos de controlo e para cada patch subdividir a superfície conforme o nível de tecelagem. Para isto foi utilizada a função auxiliar bernstein(i, n) que calcula o i-ésimo polinômio no intervalo t formando uma superfície “suave” sobre todos os pontos de controle. Para avaliar a posição de um ponto na superfície foi utilizada a função bezierPatchEval(cp, u, v) que recebendo a lista com os 16 pontos de controlo aplica a combinação dos polinômios de Bernstein nas duas direções u e v para calcular a posição do ponto na superfície.

Mudanças no engine

Nesta terceira fase o engine foi alvo do maior número de mudanças, começando pela funções que faziam parse aos ficheiros XML como readXML() e readXMLgroups() sendo que agora a transformação *rotate* podia assumir um novo elemento “time” que permite criar uma animação contínua em torno dos eixos. Da mesma forma, a transformação translate pode também assumir os novos elementos “time”, “align” e “point” onde:

- “time” define o tempo que demora a percorrer a curva
- “align” indica se o objeto deve alinhar a sua orientação com a direção da curva ao longo do percurso.
- “point” ponto de controlo da curva de Catmull-Rom

De forma a armazenar os dados de cada grupo foram adicionados 4 novos campos à definição de Group, sendo eles:

- rotationTime: guardar os atributos do “time” num rotate
- translationTime: guarda o valor do “time” de uma translação
- align: guarda um booleano
- pontosTranslacao: guarda os pontos de controlo da translação

Para as transações com o elemento “time” foi implementada a função getGlobalCatmullRomPoint() que passando por todos os vértices ia aplicando a função getCatmullRomPoint() em conjuntos de quatro vértices que calcula a matriz de coeficientes A = M * P sendo P os 4 pontos que definem o segmento e M a matriz gerada a partir dos pontos de controle escolhidos. Já com a matriz de coeficientes polinomiais definida esta é multiplicada com o polinômio cúbico T de forma a determinar a posição da curva no instante T. Para além disto é também calculada a derivada revelando a direção da orientação do ponto.

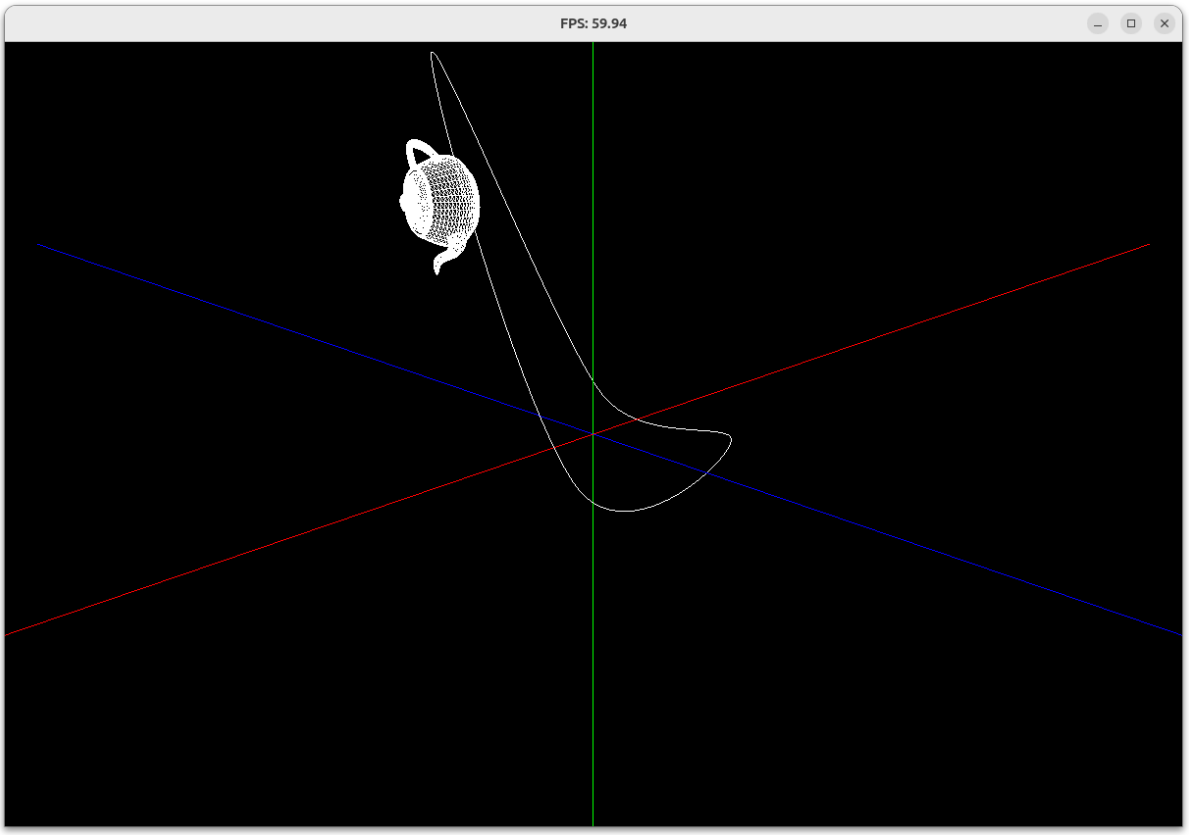
Caso o valor do elemento “align” corresponda a TRUE esse valor de derivada é utilizado para orientar o modelo ao longo da curva sendo para isso utilizadas as funções cross que calcula o vetor perpendicular entre dois vetores, normalize que garante que todos os vetores têm um comprimento igual a 1, de forma a evitar que a matriz de rotação não estique nem escolha o modelo enquanto o orienta com a direção da curva.

VBOs

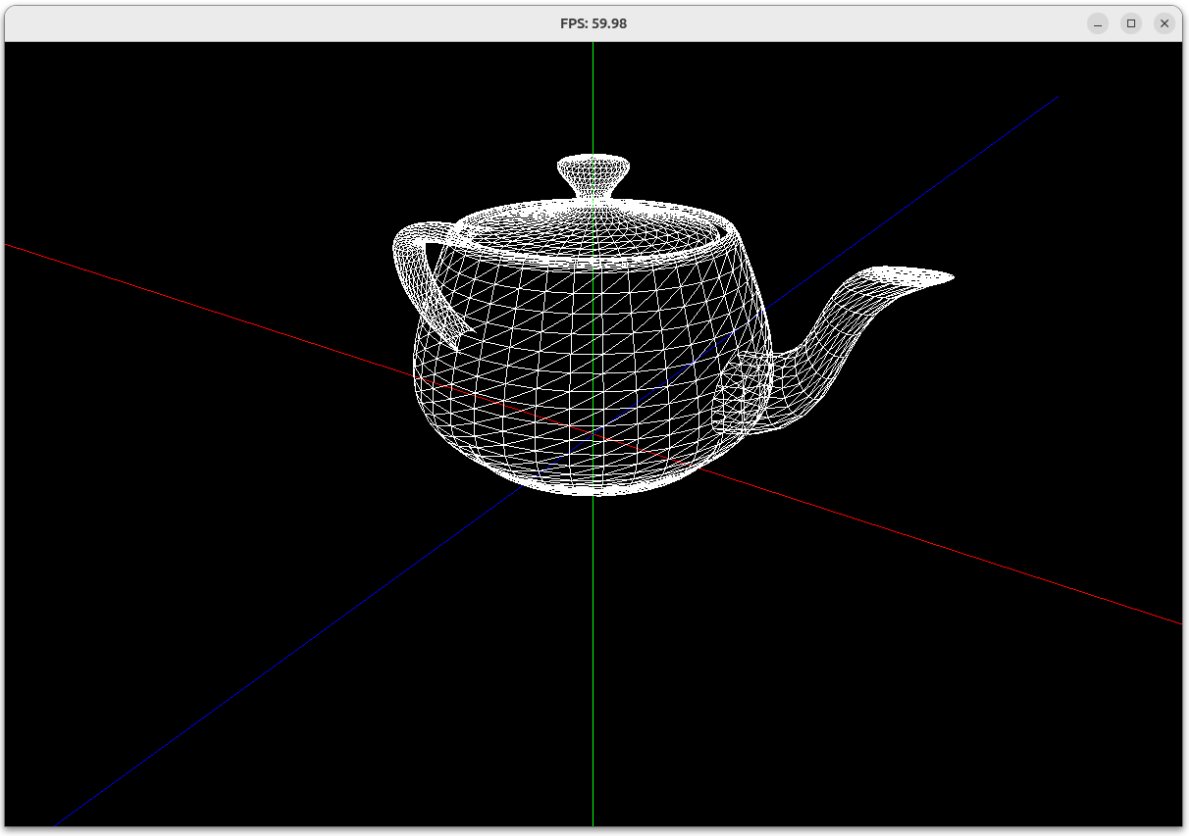
Nesta terceira fase usamos VBOs, decidimos que cada grupo tem o seu próprio VBO, ao criar o VBO, que é quando estamos a ler os ficheiros .3d(no readFile) guardamos o seu index do ARRAY BUFFER no grupo e guardamos também quantos vértices tem esse grupo.
Depois no draw, após ter as transformações prontas, chamamos o VBO com o glBindBuffer, definimos como queremos com o glVertexPointer e desenhamos com o glDrawArrays.

Demos

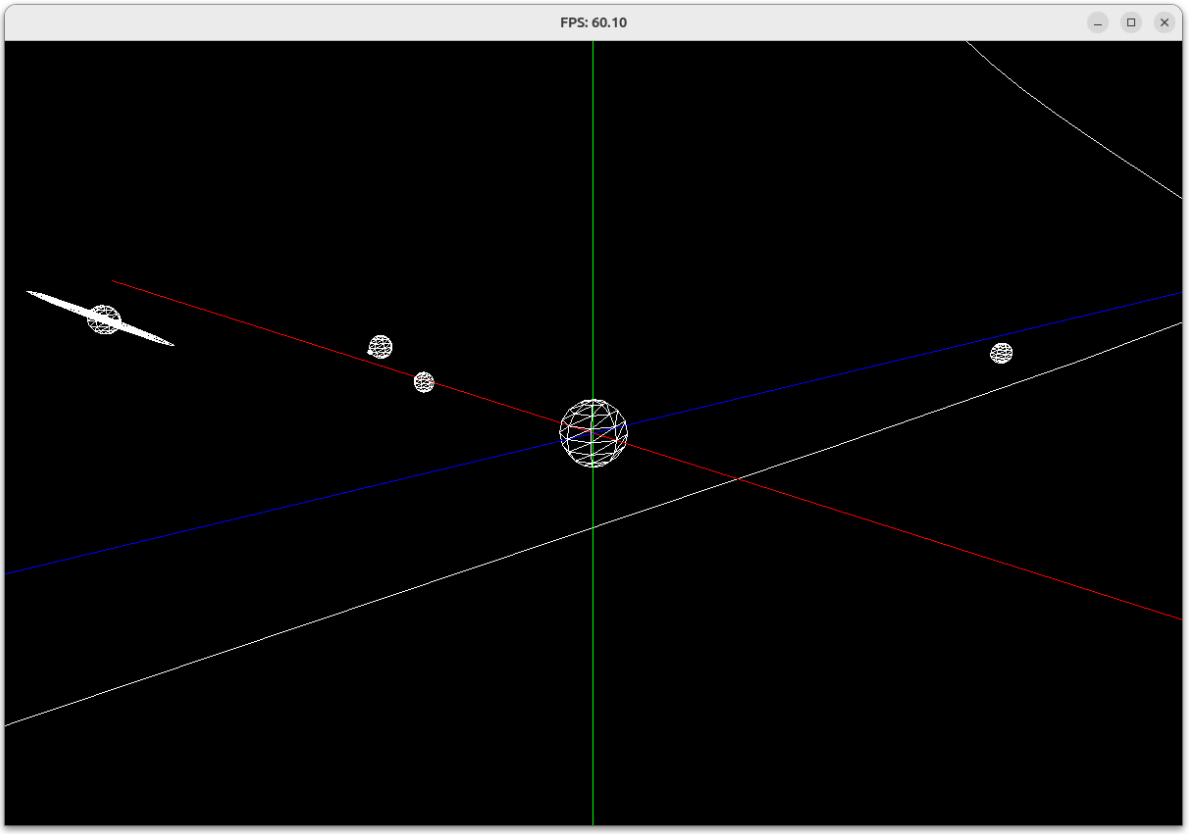
- test_3_1 : teapot a seguir a curva de CatmullRom;



- test_3_2: teapot estática;



- solar_system_dynamic : sistema solar dinâmico.



Testar

Para correr assumindo que está no diretório do projeto(ou seja no CG-main) basta fazer:

- make
- ./Generator/generator nome [argumentos do nome] nome.3d
- ./Engine/engine teste_desejado