Universidade do Minho

3º ano do Mestrado Integrado em Engenharia Informática Computação Gráfica

Ano lectivo 2016/2017

Fase 3 - Curvas, Superfícies e VBOs

AUTORES:

Diana Oliveira (a67652) Manuel Moreno (a67713) Xavier Francisco (a67725)

Braga, 2 de Maio de 2017



Conteúdo

1	Intodução	2
2	Superfícies de Bezier	2
3	Rotações e Translações	2
4	m VBOs	2
5	Análise dos Dados	3
6	Conclusão	4

1 Intodução

Esta terceira fase trata da implementação de translações definidas por pontos de uma curva e por tempo, a adaptação da rotação para ser também associada ao tempo e ainda a utilização de VBOs. O objectivo desta fase é conseguir fazer animações utilizando as superficies de Bézier.

2 Superfícies de Bezier

As superfícies de Bezier são definidas por um conjunto de pontos de controlo na direção dos quais são esticadas como se houvesse uma força atrativa.

O seguinte algoritmo explica como achar um ponto na superficie. Os argumentos **s** e **t** são incrementados linearmente de modo a fazer sampling à superficie.

3 Rotações e Translações

As rotações foram facilmente implementadas simplesmente tendo um contador com noção de tempo, para alterar o ângulo.

Pelo contrário, o de translação envolveu a implementação do algoritmo de catmull-rom. Esse algoritmo envolve encontrar pontos arbritários entre o 2º e 3º ponto de 4 pontos, no total. Percorremos os pontos todos de modo a agrupar de 4 em 4, para encontrar uma trajetória continua.

4 VBOs

Os VBOs(Vertex Buffer Objects) visam melhorar a performance do OpenGL e consistem em armazenar os triângulos e as suas informações e que depois são passadas à memória da placa para que depois esta os desenhe, libertando assim o processador de carga desnecessária.

Como podemos observar na figura em baixo, começamos por obter o caminho para o ficheiro com os vértices da figura desejada na função DrawModeVBO. De seguida damos

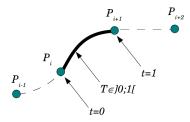


Figura 1: Spline

início a função "load Vertices" no qual é guardado em memória (utilizando um buffer) todos os vértices prontos para serem desenhados.

Após o preenchimento dos dados no buffer, dizemos como os dados estão formatados e finalmente desenhamos no ecrã.

Figura 2: VBOs

5 Análise dos Dados

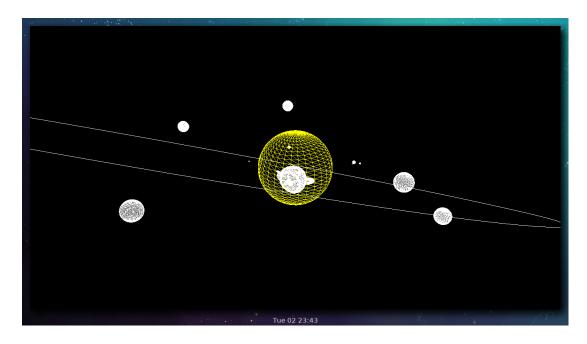


Figura 3: SolarSystem frame A

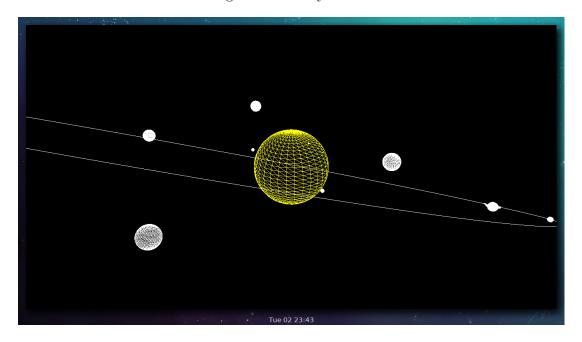


Figura 4: SolarSystem frame B

6 Conclusão

Nesta fase de trabalho foi possível aprender que o uso de VBO's reduz o tempo de renderização devido a alocar todos os vértices em memória o que os torna mais rápidos de serem lidos. Usámos curvas de Bezier e splines de Catmull rom, para criar superficies curvas a partir de pontos.

Para além disso, passámos o gerador para C++ com uma arquitectura mais própria e o

parser foi re-implementado em tiny XML2.