

Universidade Federal de Santa Maria
Curso de Ciência da Computação
Disciplina: Computação Gráfica
Primeiro semestre de 2019
Prof. Cesar Tadeu Pozzer
18/06/2019

Trabalho 4 - Curvas B-Spline 3D



Speed Racer Movie



Stunts

Descrição

Desenvolva um programa em C++, utilizando a API OpenGL, para controlar um carro sobre uma pista de corrida semelhante a um autorama. A pista deve ser formada por curvas B-Spline. O programa deve permitir que o usuário controle a velocidade e direção do carro enquanto o carro se movimenta sobre ela.

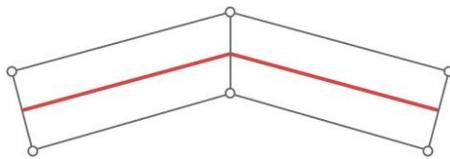
Objetivos

- Explorar o uso de vetores e suas operações
- Explorar o uso de curvas
- Explorar interação com o mouse e teclado.
- Explorar a API OpenGL

Requisitos Mínimos (Nota 9.0)

- A pista deve ser um circuito fechado, em 3D, com elevação variada.
- O usuário deve poder controlar a direção do carro usando as setas do teclado.
- A geração da pista pode ser estática – feita à mão – Apenas os pontos de controle e grafo de controle.
- A geometria do carro deve ser 3D (pode ser um cubo). Pode-se escolher visão interna ou externa do carro (gluLookAt).
- Permitir que o usuário visualize o vetor direção do carro. Crie um botão para ativar e desativar essa funcionalidade.
- Implementar qualquer tipo de iluminação (no mínimo difusa).
- O carro deve acompanhar a elevação da pista.
- O programa deve ter controle de FPS.

- Programa expande a linha definida pelo usuário e gera a pista com largura específica (gerar as linhas pretas a partir da linha vermelha).



Bônus (valores máximos)

- Fazer pistas com inclinação nas curvas (1 pontos)
- Permitir que o usuário gere a pista utilizando o mouse. (1 ponto)
- Utilizar textura (1 ponto)
- Utilizar modelos 3D elaborados – usar algum loader de objetos 3D (1 ponto)
- Carro sai da pista se sua velocidade for muita alta em curvas fechadas (2 pontos).
- Pistas com loop (2 pontos)
- Controle da física do carro – suspensão, atrito, gravidade (2 pontos).
- Rampas e saltos (2 pontos)
- Adicionar viewports para mostrar diferentes visões do carro (1 ponto).
- Replay da corrida vista sob qualquer ângulo (1 ponto)
- Etc.

Data e Formato de Entrega

- Data: 8/julho/2019, horário de expediente. O trabalho deve ser **apresentado** individualmente. Após esse período o trabalho será desconsiderado.
- No email e no cabeçalho do arquivo, devem conter o nome completo e matrícula do aluno.
- O arquivo deve ser enviado para pozzer3@gmail.com com o subject "CG T4".
- Deve-se enviar fontes e o projeto para a IDE **Code::blocks**. Envie **apenas** os arquivos **de projeto, código fonte e modelos**.
- O programa deve ser enviado em um arquivo compactado fulano.rar (fulano = login ou nome do aluno). Dentro deste arquivo deve haver um diretório com o mesmo nome do arquivo e dentro deste diretório os arquivos do trabalho.
- Ex: o arquivo pozzer.rar deve conter um diretório chamado pozzer, e dentro do diretório devem estar os arquivos do trabalho.

Critério de Avaliação

- Documentação: descrever no cabeçalho de cada arquivo a ideia geral do código e detalhes específicos de partes que mereçam uma explicação – não comente por exemplo o que faz b++.
- README.txt: incluir um arquivo "README.txt" contendo informações sobre quais funcionalidades foram implementadas (requisitos e extras).
- Pontualidade: Trabalhos não entregues na data não serão avaliados e receberão nota zero.
- Legibilidade: nome de variáveis, estruturação do código. O código digital a ser entregue deve ter 4 espaços de identação e não deve possuir tabulações.
- Clareza: facilidade de compreensão – evite códigos complexos e desnecessários. Adote a solução mais simples possível.
- Funcionalidade: o programa deve satisfazer todos os requisitos. Programas que não compilarem ou que não atenderem nenhum requisito receberão nota 0 (zero).

Você pode discutir estratégias e ajudar o colega na implementação, porém evite passar código fonte. Programas semelhantes terão a nota 0 (zero).