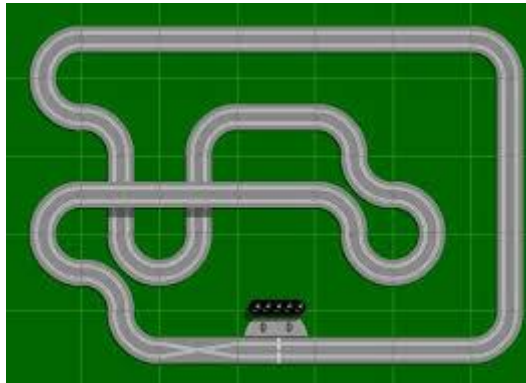


Trabalho 2 - Curvas: Bézier e B-Spline



Descrição

Desenvolva um programa em C++, utilizando a **API SCV 5.0**, para controlar um carro sobre uma pista de corrida semelhante a um autorama. A pista deve ser formada por curvas Bézier, B-Spline ou NURBS. O programa deve permitir que o usuário gere a pista através do mouse e depois controle a velocidade 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.

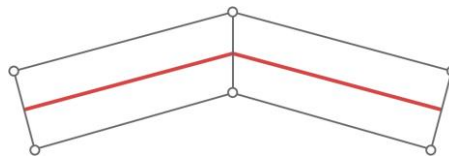
Requisitos Mínimos

- Implementar uma das seguintes curvas: Bézier, B-Spline ou NURBS. As curvas devem ter no mínimo grau 3.
- Continuidade mínima deve ser C1.
- A pista deve ser um circuito fechado.
- Permitir que o usuário gere a pista utilizando o mouse. Dica: você pode se basear nesse vídeo: <https://www.youtube.com/watch?v=1HCbe8yTHNw>
- Deve-se aplicar transformações sobre o carro para que ele ande sobre a curva. A geometria do carro não pode ser um círculo.
- Permitir que o usuário visualize grafo de controle da(s) curva(s) e o vetor direção do carro. Crie um botão para ativar e desativar essa funcionalidade.
- Renomear as classes geradas pelo FreeFormInterfaceDesigner com nomes intuitivos!

- Deletar o código das callbacks não utilizadas! Deletar as classes não utilizadas!

Bônus

- Fazer com que o carro mantenha uma velocidade constante ao percorrer as curvas.
- Implementar mais de um tipo de curva e permitir que o usuário escolha qual tipo usar.
- Desenhar a pista de maneira que sejam geradas mais amostras onde a curva é mais "fechada". Dica: http://antigrain.com/research/adaptive_bezier/.
- Programa expande a linha definida pelo usuário e gera estrada com largura específica.



- Utilização de sprites (imagens).
- Baseado na(s) curva(s), gerar pista com mesh (malha de triângulos).
- Carro sai da pista se sua velocidade for muita alta em curvas fechadas.
- Etc.

Data e Formato de Entrega

- Data: 18/maio/2015, 23:59h.
- No email e no cabeçalho do arquivo, devem conter o nome completo e matricula do aluno.
- O arquivo deve ser enviado para pozzer3@gmail.com, gcardozo@inf.ufsm.br e afrasson@inf.ufsm.br com o subject "CG T2".
- Deve-se enviar fontes e o projeto para o compilador **Visual Studio 2013**. Envie **apenas** os arquivos **.sln** e **.vcxproj** juntamente com a pasta **src**.
- 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 indentação 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).