

## Trabalho 3 – Visualização de Objetos em 3D

### Descrição:

Implemente um programa em C++ para fazer a modelagem e visualização de um objeto 3D, representado por meio de sweep, por meio de uma câmera sintética. Deve-se criar uma interface **amigável** de interação (mouse, teclado, menus, etc). A visualização deve ser por meio de projeção perspectiva e ortográfica (selecionável).

O trabalho tem vários níveis de complexidade, e são incrementais:

- 1) Fazer a geração de um objeto por sweep rotacional, onde o usuário pode clicar com o mouse para definir a forma do objeto. Com isso pode-se gerar objetos cilíndricos como copos, vasos, etc. Deve-se poder visualizar usando pontos ou linhas, com opção para projeção ortográfica e perspectiva (wireframe) – Nota 9
- 2) Fazer também a geração de uma mola, também por sweep. O usuário deve definir como é a forma da seção transversal da mola. Deve-se poder visualizar usando pontos ou linhas (wireframe) – Nota 10
- 3) Implementar uma câmera sintética (modelo 3 visto na apostila – 1 ponto e 3 vetores). A câmera deve ter 6 graus de liberdade: 3 de translação e 3 de rotação. As funções de câmera devem estar em funções em claras e definidas. Vejam <http://www.opengl.org/sdk/docs/man/xhtml/gluLookAt.xml> para mais detalhes. (1 ponto de Bônus)
- 4) Trocar a malha plana por uma superfície animada de Bézier ou B-Spline. Deve-se usar uma matriz com no mínimo 10x10 pontos de controle. Deve-se fazer a triangulação da malha. Deve-se garantir continuidade C1 ou superior entre os patches que gerarão a superfície. Os pontos de controle podem ser gerados de forma semi-aleatória ou por meio de uma imagem (height map). A malha do terreno deve ser triangular. Defina uma estrutura de dados para gerar a malha. Pode-se aplicar transformações nos pontos de controle para simular uma malha em movimento (como se fosse a superfície de um oceano). (2 ponto de Bônus)
- 5) Aplicar sobre os objetos gerados alguma forma de preenchimento de polígonos com iluminação. Não pode ser usada a API OpenGL. (5 pontos de bônus)
- 6) Etc. (x pontos de bônus)

Deve-se utilizar a API SCV 4.2 (**versão nova**) para implementar o trabalho. <http://www-usr.inf.ufsm.br/~pozzer/scv/>

**Dica:** faça inicialmente a visualização com projeção em perspectiva de um cubo centrado na origem. **Faça a projeção sem o uso de matrizes de transformação.** Crie uma função `void projeta(float x, float y, float z, int *xt, int *yt)` **que retorna as coordenadas de tela  $x_t$  e  $y_t$  para cada ponto dado em função do centro de projeção da câmera.** Assuma a existência de 1 ponto de fuga. Quando isso funcionar, comece a implementar o resto.

#### Data e Formato de Entrega:

- Data: 13/06/2012.
- 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](mailto:pozzer3@gmail.com) e [lquatin@inf.ufsm.br](mailto:lquatin@inf.ufsm.br) com o *subject* “CG T3”. Deve-se enviar fontes e o projeto para o Microsoft Visual Studio 10.
- 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. Deve-se enviar somente os fontes e projeto Visual Studio 2010 (disponível na pasta userProject do SCV). **É proibido** o envio de arquivos **pdb, ilk, sdf, idb, obj, exe, etc.** **Envie somente o que for necessário para compilação.** Não deve ser enviada a lib do SCV.

#### 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++`.
- **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 **3 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).