

## Trabalho 4 – Simulador Virtual 3D de uma Escavadeira

### Descrição:

Implemente um programa em C++/OpenGL (opcional uso da API SCV), que simula o funcionamento de uma retro-escavadeira. Vídeos demonstrativos do funcionamento podem ser encontrados no youtube.

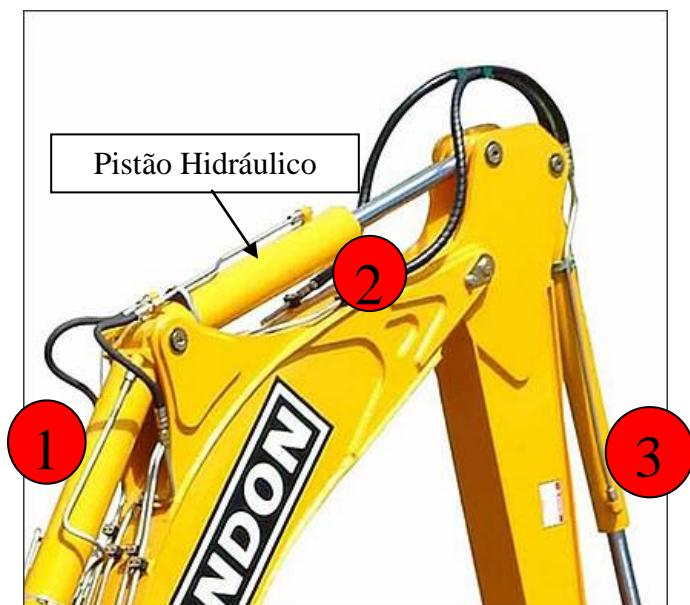
<http://www.youtube.com/watch?v=HbdqYnSPi9Q&feature=fvw>

<http://www.youtube.com/watch?v=Bmf2AJ0C0gs&feature=related>

Deve ser modelado e implementando o funcionamento do **braço traseiro**. Todo o restante da máquina pode ser modelado por um único cubo.



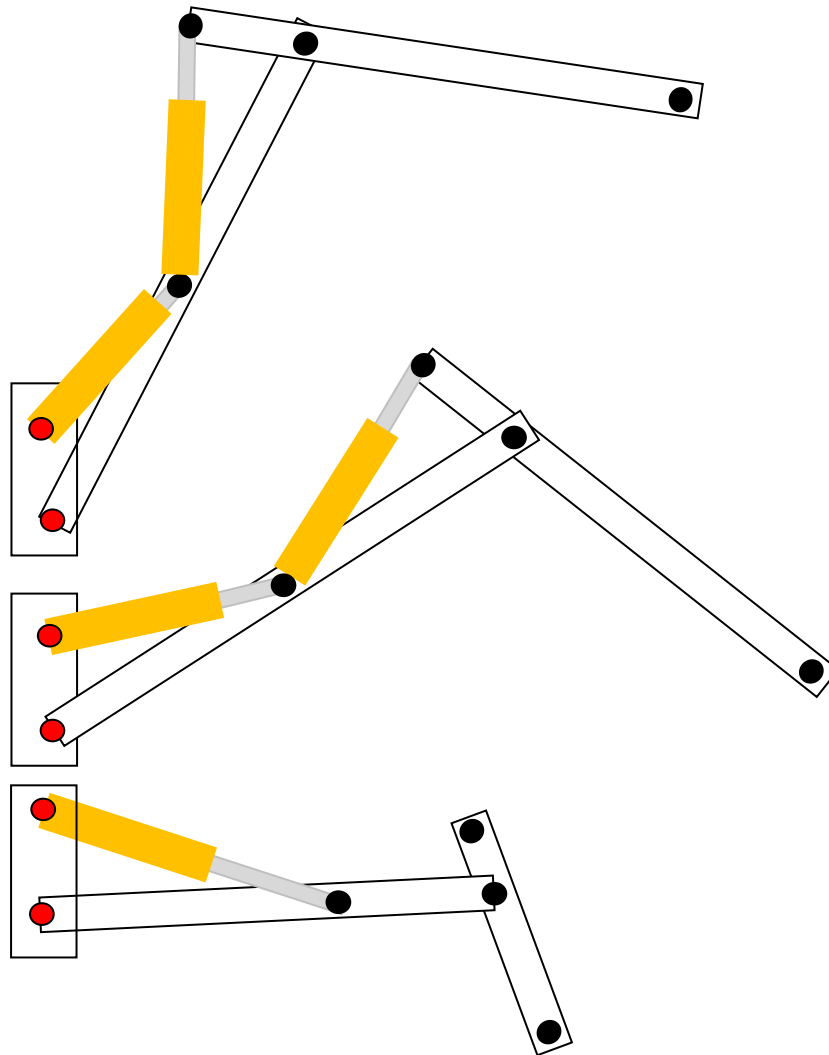
O braço traseiro deve ser modelado como nas seguintes figuras e deve ter 3 pistões hidráulicos. Deve-se fazer uso de vetores em sua modelagem. O **principal** é a modelagem do pistão (que estica e encolhe - semelhante a uma seringa) e de sua movimentação atrelada ao braço. Deve existir uma classe pistão na aplicação. Deve-se deixar bem destacado os pontos que conectam os pistões com os braços.



O braço como um todo pode girar ao redor do eixo de rotação, como mostrado na seguinte figura.



Deve-se observar que quando o braço se move, mudam os ângulos de direção do pistão, como mostrado de forma genérica nas seguintes figuras. Os pontos de apoio móveis dos pistões (pontos pretos) devem ser calculados com matrizes de rotação. Os pontos vermelhos são pontos fixos.



Na escavadeira devem estar presentes os seguintes itens:

- O braço da escavadeira deve ser composto por 3 partes móveis (braço, antebraço e concha), como mostrado nas figuras. Crie meios de interação **intuitivos** com mouse e/ou **teclado** para operar o braço mecânico. Deve-se exibir os pistões hidráulicos conectados a cada parte, como nas figuras. Eles devem ser movidos (alongados ou encurtados) juntamente com o braço a que estiverem conectados. Também devem ser utilizados como limitadores de curso de movimentação dos braços (como ocorre no mundo real).
- A geometria utilizada pode ser bem simples.
- Deve-se poder definir a localização da câmera sintética: dentro da cabine ou externa.
- Se locomover.

Pode ser usado ferramentas para fazer o carregamento de modelos 3D. Mas os modelos tem que estar relacionados ao braço/escavadeira, ou seja, não conta carregar uma casa para deixar o ambiente mais bonito. A modelagem da escavadeira pode ser feita em ferramentas de modelagem, como o 3DMax por exemplo.

## Objetivos do trabalho

- Praticar o uso do OpenGL em aplicações reais
- Uso de vetores e suas operações
- Hierarquia de movimentação (push e pop matrix)

## Tópicos adicionais:

- Recursos de iluminação (veja demo em CGA).
- Adição de texturas
- Adição de física
- Interação do braço com objetos do cenário.
- Pode-se exibir forças em cada junta ou pistão de movimentação do braço da escavadeira.

## Pontuação do Trabalho

- Implementar 1 braço móvel com um pistão (Figura acima). O Cálculo do ponto de apoio deve ser calculado corretamente. Deve-se fazer um braço que possa ser facilmente demonstrado que o cálculo esteja correto – Nota 8
- Implementar 3 braços móveis, inclusive com algo se se pareça com a concha – Nota 11
- Simular o todos os movimentos de um braço real da escavadeira e adicionar um pouco de realismo (Geometria detalhada, iluminação, etc) – Bônus 3 pontos
- Simular também a movimentação da escavadeira (em curvas) de forma realista, com pneus – Bônus 3 pontos.
- Etc – Bônus variável

## Dicas para o Trabalho

- Defina uma classe para desenhar cilindros. Utilize o conceito de sweep rotacional (função paramétrica) para definir os vértices. Utilize GL\_QUAD\_STRIP para a lateral e GL\_TRIANGLE\_FAN para as extremidades. Se for utilizada iluminação, esta classe já deve definir as normais.
- Defina uma classe para desenhar os braços (podem ser quadrados). Utilize os mesmos conceitos para gerar os cilindros.
- Defina uma classe para gerenciar um pistão. O pistão deve poder encolher e esticar, respeitando os limites. Para desenhar o pistão, utilize dois cilindros: o interno e o externo. Cada cilindro deve ter uma cor diferenciada. Se for usada iluminação, o cilindro interno deve ter reflexão especular, e o externo somente difusa. Para desenhar cada pistão, deve-se informar apenas 2 pontos no espaço. Esses pontos devem ser calculados com matrizes de transformação.
- Aplique a transformação de rotação ao braço, por exemplo, e após calcule os pontos de apoio, usando os mesmos parâmetros de rotação. Somente após, renderize os pistões. O processo contrário, ou seja, esticar o pistão e definir onde o braço vai parar é muito mais complicado.
- Implemente cada parte separadamente e teste. Após, integre ao projeto final.

## Data e Formato de Entrega:

- Data: 6/07/2012. O trabalho deve **ser apresentado** até a data limite. Agendar horário.

- 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 T4”. 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).