

Universidade Federal de Santa Maria
Curso de Ciência da Computação
Disciplina: Computação Gráfica
Primeiro Semestre de 2012
Prof. Cesar Tadeu Pozzer
Data: 19/04/2012

Trabalho 2- Transformações Geométricas

Descrição:

O jogo **Gorillas** consiste em disparar um objeto (uma banana) contra um alvo. O jogo pode ser encontrado no seguinte site: <http://www.kongregate.com/games/Moly/gorillas-bas>. Jogo esse jogo para compreender bem o seu funcionamento e para ter ideias de implementação.

Implemente esse jogo em C++, fazendo uso da API SCV, em que o jogador controla um canhão (ou o que preferir) e deve atirar projéteis. O projétil deve ter o formato de uma seta (→). O jogador deve poder realizar a rotação do canhão, além de possibilitar ao usuário modificar a velocidade com que o canhão dispara o projétil. As seguintes figuras foram retiradas do link acima e ilustram as definições acima citadas



O projétil a ser lançado deve ilustrar a trajetória que está seguindo, por isso **deve ter a forma de uma seta** (não utilize um círculo como projétil, visto que este não terá como ilustrar a direção que o projétil seguirá). **Deve-se utilizar concatenação de matrizes de transformação, bem como trabalhar com vetores e suas propriedades.**

Deve-se tratar a colisão do projétil com o alvo e contra os prédios.

Critérios que serão avaliados:

- Classes em C++ para definição de vetores e transformações geométricas
- Sistema do controle do canhão
- Sistema de movimentação do projétil. Utilize o mouse para isso.

Critérios básicos de interação:

- Controle da direção do canhão com o uso do mouse
- Controle da força de disparo
- O disparo deve ser com o clique do botão do mouse.

Critérios avançados de interação (Bônus):

- Colisão do projétil com os alvos
- Implementação de conceitos básicos de física para definição da trajetória do projétil.
- Tipos diferentes de tiros (efeitos especiais, pesos diferentes, etc.)
- Tipos diferentes de alvos (alvos móveis, alvos que mudam trajetória do projétil, etc.)
- Inclusão de obstáculos, como ocorre no jogo
- Base do canhão móvel
- Criação de várias fases do jogo
- Algoritmo de geração de cenários
- Adicionar uma IA básica ao jogo (jogador inimigo – em turnos, igual ao jogo citado)
- Tratar velocidade do vento
- Realizar interação com o ambiente (destruir parte do prédio a cada tiro. Para exemplificar, veja o funcionamento do jogo do link acima)
- Etc.

O trabalho deve apresentar uma **lista de instruções**, explicando de forma como o usuário deve interagir com o programa. Enumere no início do código fonte (arquivo main.cpp) os quesitos que foram implementados.

O trabalho deve ser individual. Dúvidas entrem em contato com o monitor (**Leonardo Quatrin Campagnolo**) ou comigo. A avaliação também poderá ser oral (a critério do professor).

Data e Formato de Entrega:

- Data: 10/05/2012.
- 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 e lquatrin@inf.ufsm.br com o *subject* “CG T1”. 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.

- **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 idéia 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).