

Universidade Federal de Santa Maria
Curso de Ciência da Computação
Disciplina: Computação Gráfica Avançada
Segundo Semestre de 2016
Prof. Cesar Tadeu Pozzer
Data: 21/09/2016

Trabalho 2 – Shader

Desenvolva um programa em C++, utilizando a API OpenGL 4.x que carrega uma textura. Ao mover o mouse sobre a textura, deve-se dar zoom nas proximidades do mouse (raio de atuação), simulando uma luneta de um rifle. A área que estiver fora do raio de atuação do mouse deve ser desfocada.

Deve-se simular um efeito de tremor, como acontece quando se está mirando em um alvo com uma arma. Quando maior for o zoom, maior deve ser o tremor simulado.
Simular também o disparo da arma e marcar na textura a posição onde o disparo atingiria. O disparo da arma implica em mover arma para cima (recoil).



Recursos que devem ser processados em tempo real. Pode-se utilizar o scroll do mouse, juntamente com outra tecla pressionada.

- alteração do raio de atuação
- nível de zoom da lupa
- etc (ex: distorções, efeitos, etc).

Extras:

1. Renderizar uma cena 3D para textura (render to texture) e então usar essa textura para dar zoom. Modelar um cenário 3D com personagens (inimigos do mal) e poder acertar neles.
2. Etc.

Data e Formato de Entrega

- Data: 26/set/2016, 23:59h (prox segunda-feira).
- No email e no cabeçalho do arquivo devem conter o nome completo do aluno.
- O arquivo deve ser enviado para pozzer3@gmail.com, mdalcin@inf.ufsm.br com o subject "CGA T2".
- Deve-se enviar fontes e o projeto para o compilador **Visual Studio 2015**. Envie **apenas** os arquivos **de projeto, código fonte e modelos/texturas**.
- 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 nota 0 (zero).