Universidade Federal de Mato Grosso do Sul
Faculdade de Computação
Disciplina: Computação Gráfica
Professor: Anderson V. de Araujo

Trabalho Prático

Descrição

Desenvolver uma aplicação em qualquer uma das linguagens usadas na disciplina (Java, C++ e Python) que atenda os seguintes requisitos:

- Adicionar e remover (uma ou mais) formas:
  - Cubo, esfera (icosphere), cone e torus;
- Habilitar e desabilitar modelos de iluminação
  - Especular, difusa e ambiente;
- Adicionar e remover fontes de luz (pontuais);
- Definir tipo de projeção e seus parâmetros associados:
  - Ortográfica ou perspectiva;
- Câmera;
  - Definir vetor de visão e posição da câmera;
- Transformar as formas:
  - Translação, Rotação, Escala e *Shear*.
- Mostrar/Esconder os eixos;
- Salvar a imagem corrente em arquivo PNG;
- Mostrar as luzes posicionadas na cena;
- Ler comandos de entrada padrão por linha de comando.

Informações sobre a Implementação

- As formas adicionadas inicialmente devem estar na posição (0,0,0) e conter as seguintes dimensões – opções do blender: cube (diagonais=1,1,1 e -1,-1,-1), sphere (icosphere, 5 subdivisões, size=1), cone (vértices =32, raio1=1, raio2=0, profundidade=2) e torus (major seg=48, minor seg=12, major radius=1, minor radius=0.25) – Exportar no Blender no modo smooth
- A cada execução de um novo comando, a imagem sendo renderizada deve ser atualizada;
- É obrigatório ler comandos descritos na tabela anterior de entrada padrão por linha de comando. Caso alguma funcionalidade não tenha sido implementada, responda com a mensagem “NOT IMPLEMENTED”;
- A câmera inicialmente deve estar na posição (0, 0, 0); O *lookat* (0, 0, -1) e o vetor *up* (0, 1, 0). Parâmetros para *ortho* (-1f, 1f, -1f, 1f, 1f, -1f) e *frustum*: (-1f, 1f, -1f, 1f, 0.5f, 2f). Projeção padrão: *ortho*;
- Shading padrão: *smooth;*
- Parâmetros para a reflexão specular n = 32;
- Podem ser utilizadas bibliotecas externas desde que sejam apresentadas no relatório junto com o link para download e com informações sobre ela;
- Resolução da tela de desenho deve ser de 640x480;
- Qualquer parâmetro ou valor não definido neste documento pode ser fixo e deve ser escolhido pelo aluno desde que explicado no relatório;
- Todas as luzes devem ser brancas e pontuais, máximo 10 inseridas. O cálculo da intensidade de luz deve usar a atenuação de acordo com a posição da luz na cena. Parâmetros para a atenuação: a_0=1, a_1=1 e a_2=0.25;
- Cor de fundo padrão deve ser preta e de desenho inicial branca;

- **É obrigatória a implementação usando a API Programável da OpenGL (3.0+)** – usando *shaders* na GLSL;
- Não é necessária nenhuma interação com o mouse ou teclado, mas pode ser implementada.

Tabela de comandos de entrada

Comando	Descrição
`add_shape shape name1`	Adiciona a forma correspondente de tamanho centrada na origem com o nome `name1` a cena. Formas possíveis: `cube`, `sphere`, `cone` e `torus`. Ex.: `add_shape cube cube1`
`remove_shape name1`	Remove a forma com o nome `name1` da cena. Ex.: `remove_shape cube1`
`add_light name1 float float float`	Adiciona uma nova luz na posição (x, y, z) na cena. Máximo 10. Ex. `add_light light1 0.5 0.3 1.0`
`remove_light name1`	Remove a luz com o nome `name1` da cena. Ex.: `remove_light light1`
`reflection_on type float`	Habilita o tipo de reflexão da luz nos objetos da cena e define o coeficiente de reflexão associado (k). Coeficiente de brilho da reflexão especular fixo. Tipos de reflexão possível: `specular`, `diffuse` e `ambient`. Ex.: `reflection_on specular 0.3`
`reflection_off type`	Desabilita o tipo de reflexão da luz nos objetos da cena. Tipos de reflexão possível: `specular`, `diffuse` e `ambient`. Ex.: `reflection_off diffuse`
`shading type`	Define o tipo de shading usado para renderizar a cena. Tipos de shading possíveis: `flat`, `smooth` e `phong`. Ex.: `shading flat`
`projection type float float float float float float`	Define o tipo de projeção. `type` pode ser: `ortho` ou `perspective`. Tamanho do *frustum* definidos pelos parâmetros (left, right, bottom, top, nearVal, farVal) Ex.: `projection perspective -1 1 -1 1 0.5 2`
`translate name1 float float float`	Translada a forma correspondente de acordo com os três parâmetros em (x, y, z). Ex.: `translate cube1 1.0 0.0 0.0`
`scale name1 float float float`	Redimensiona a forma correspondente de acordo com os três parâmetros em (x, y, z). Ex.: `scale cube1 0.5 0.5 0.5`
`shear name1 float float float`	*Inclina* a forma correspondente de acordo com os três parâmetros em (x, y, z). Ex.: `shear cube1 0.5 0.5 0.5`
`rotate name1 float float float float`	Rotaciona a forma correspondente de acordo com o ângulo e o vetor (x, y, z). Ex.: `rotate cube1 45 1.0 0.0 0.0`
`lookat float float float`	Faz a câmera olhar para o ponto definido pelos parâmetros (x,y,z). Ex.: `lookat 0.0 0.0 0.0`
`cam float float float`	Define a posição da câmera por meio dos parâmetros (x,y,z). Ex.: `cam 0.0 0.0 0.0`
`color name1 float float float`	Define a cor da forma com nome name1 usando três parâmetros (r, g, b). Ex.: `color cube1 1.0 0.0 0.0`
`axis`	Mostra ou esconde os eixos x, y e z. x: Eixo x vermelho, y verde e z azul.
`save filename`	Salva a visualização corrente em uma imagem png com o nome `filename`. Ex.: `save test.png`
`lights`	Mostra/esconde as luzes posicionadas na cena (um ponto amarelo para cada luz).
`wire_on`	Mostra todos os objetos no formato wireframe (cor das linhas deve ser a cor do objeto).
`wire_off`	Mostra todos os objetos no formato normal (não wireframe).
`quit`	Sai do programa.

Relatório

Entregar um **relatório em PDF** apresentando:

- Nomes dos alunos;
- Linguagem de programação e versão;
- Versão da OpenGL e GLSL;
- Parâmetros fixos (constantes) utilizados na implementação;
- Informações sobre os arquivos no zip (código do projeto enviado);
- Funções (implementações) adicionais da aplicação,
- Outras informações que achar pertinente.

Não entregar o relatório impresso. O relatório deve **obrigatoriamente** ter entre 1 e 5 páginas **no total**. **Não deve** ter capa, contracapa, bibliografia ou índice.

Entrega

Entregar o projeto zipado com todos os códigos desenvolvidos e o PDF do relatório. O projeto deve ser desenvolvido **em dupla ou individualmente**. No caso da dupla, a nota vai ser a mesma para ambos os alunos. Basta que apenas um aluno da dupla faça a entrega no AVA. A data de entrega será definida no AVA.

Avaliação

A avaliação do trabalho vai ser realizada de acordo com a implementação e o relatório entregue. No total a nota do trabalho é no máximo 10 (verificar o peso do trabalho na nota final no SISCAD). Todo o código deve ser comentado.

Se eu desconfiar que o trabalho foi copiado de outro colega ou da internet (ou de alguma maneira não foi você que fez) eu vou convidá-lo a me explicar o código na minha sala. Em caso de confirmação, é zero para o trabalho.

Quaisquer dúvidas no desenvolvimento do trabalho devem ser enviadas para o e-mail **andvicoso@facom.ufms.br** ou pessoalmente.

Bom trabalho!