

## Lab 2

### Primitivas e transformações

Para você resolver este lab, você receberá um conjunto de arquivos fonte a partir dos quais você deve desenvolver a sua solução. O objetivo é que você evolua o código entregue de modo que:

- 1 - As primitivas estudadas sejam implementadas;
- 2 - O sistema de carregamento e exibição de figuras esteja integrado;
- 3 - As transformações estudadas estejam integradas.

*Para resolver o problema 1:*

- a - Crie as primitivas que herdam de Objeto2D. Pode ser necessário reimplementar o método display; e
- b - Implemente retas (algoritmo de Bresenham) e curvas (Bézier até o cúbico).

*Para resolver o problema 2:*

- a - Utilize o método "carregarArquivo" da classe "LerForma".

*Para resolver o problema 3:*

- a - Herde da classe "*Transformações*" para cada nova transformação **nas 3 dimensões** (repare que a classe Objeto2D, na verdade, implementa utilizando 3 coordenadas);
- b - Utilize sempre **matrizes** para implementar as transformações. Não é permitido o uso de NENHUMA função pronta do OpenGL relativa a transformações;
- c - Para simplificar, **a composição de transformações** poderá ser feita em **chamados consecutivos** do método "transformacao" da classe Objeto2D;
- d - No construtor das classes de transformação criadas, deve ser passado o mínimo possível de parâmetros; e
- e - As combinações para as quais pode-se deduzir um caso base (exemplo: escala em relação a um ponto), devem ser implementadas em uma única matriz a partir da sobrecarga do construtor.

#### **A entrega:**

Basta enviar as classes criadas compactadas em um arquivo .zip. **NÃO ENVIE O CÓDIGO QUE VOCÊ RECEBEU!** Os nomes dos arquivos (.cpp e .h) e suas respectivas classes serão: Bezier, Reta, Escala, Rotacao, Translacao, Reflexao, Cisalhamento.

Você pode ter outras classes auxiliares às descritas acima. Caso seja o seu caso, não esquece de enviá-las também!

A avaliação dos resultados será realizada da seguinte forma:

- a - Será utilizado um arquivo, contendo a descrição dos objetos que compõem a cena, o qual será carregado a partir da função carregarArquivo;

b - Um conjunto de operações será realizado sobre estes dados.

Receberá 10 os trabalhos que apresentarem comportamento idêntico ao esperado.

Na aula do dia 19/12, alguns alunos serão sorteados para explicar partes específicas dos códigos enviados. A qualidade da explicação ponderará a nota recebida em um peso que variará entre 0 e 1.

O código precisa ser enviado até o dia 19/12 às 10h! Não há possibilidade de atrasos, já que a avaliação será no dia 19/12 às 14h.

**Você receberá:**

exemploArquivo.pcg - Um exemplo funcional de como as formas serão descritas e carregadas.

LerForma.(cpp,h) - Classe já pronta que carrega as formas e retorna um vector<Objeto2D>.

Matmult.(cpp,h) - Função para multiplicar matrizes! (Pode conter bugs)

Transformacao.(cpp,h) - Classe base para uma transformação. Possui como atributo a matriz de transformação e um método "transformar", o qual recebe um vetor de entradas e retorna a matriz resultante.

Objeto2D.(cpp,h) - Classe base de um objeto. Permite adicionar vértices, transformá-los (passando um objeto Transformação) e exibir o objeto (método display).

Vertice.(cpp,h) - Uma primitiva de lambuja. Óbvio que não poderia herdar de Objeto2D.

P.S.: - A "Engenharia" do sistema pode não estar boa. Utilize esta engenharia mesmo assim, uma vez que o seu código será integrado ao meu. Logo, não altere nada no código fornecido. Caso encontre um erro, avise para que toda a turma receba o trecho corrigido.

**Bom trabalho!**