

## 1 Manipulador de arquivos obj (Trabalho 2)

Você deve construir 3 aplicativos para manipular arquivos .obj, respectivamente chamados de SimplifyObj, ChopObj e Obj2Stl. Cada um destes programas deve ler um arquivo .obj e armazenar os vértices do objeto em uma lista de vértices e as faces do objeto em uma lista de faces.

- Ao executar o programa SimplifyObj passando como entrada um arquivo .obj, este programa deve gerar como arquivo de saída um arquivo .obj que contenha o objeto do arquivo de entrada, exceto que contendo 20% menos faces. As faces que devem ser removidas são as 20% menores faces (ou seja, faces com menor superfície do objeto original). **Os vértices que não fizerem parte de nenhuma face também devem ser removidos.**
- Ao executar o programa ChopObj passando como entrada um arquivo .obj, este programa deve gerar como arquivo de saída um arquivo .obj que contenha o objeto do arquivo de entrada, exceto que eliminando todas as faces que estejam contidas completamente na metade de cima do objeto. Para determinar quais faces estão contidas na metade de cima do objeto, calcule o ponto médio  $(x_m, y_m, z_m)$  entre o vértice de menor coordenada  $z$  e o vértice de maior coordenada  $z$ . As faces que devem ser eliminadas devem ser àquelas que tem todos os seus vértices acima deste ponto médio, ou seja, com coordenada  $z$  maiores do que  $z_m$ .
- Ao executar o programa obj2stl passando como entrada um arquivo .obj, este programa deve gerar como arquivo de saída um arquivo .stl do tipo binário que contenha exatamente o mesmo objeto do arquivo de entrada (ou seja, seu programa é um conversor do formato obj para o formato stl). Para simplificar, você pode assumir que o arquivo de entrada contém apenas triângulos como faces.

**Importante:** Além dos vértices e faces, o formato OBJ também permite definir linhas, pontos, normais, texturas e outros elementos importantes em um modelo 3D. Neste projeto, somente precisam ser tratadas as declarações de vértices e faces. As demais podem ser ignoradas. Outra simplificação é que você pode assumir que os modelos poligonais são triangularizados. **Ou seja, se o arquivo de entrada possuir alguma face que não seja um triângulo, o programa deve avisar o usuário e abortar a execução.**

## 2 Calculador de fatia para impressão 3D (Trabalho 3)

Em aplicativos de impressão 3D é comum que dado como entrada um arquivo .stl, o software deve calcular várias fatias do objeto em questão enviar para a impressora comandos para que cada uma destas fatias seja impressa. Neste trabalho você deve lidar com uma versão simplificada deste problema. Você irá calcular apenas uma fatia do objeto e gerar um arquivo de saída que indica a forma desta fatia.

Você deve construir um programa que toma como entrada um arquivo .stl e calcula a fatia que corta o objeto exatamente ao meio. Para resolver este problema, você deve encontrar todos os triângulos que sejam cortados pelo plano horizontal que corta o objeto ao meio. Este plano é o plano que está equidistante do vértice com a maior e o vértice com a menor coordenada  $z$  do objeto. Uma vez encontrados estes triângulos, você deve calcular o segmento de reta obtido pela intersecção de cada triângulo com o plano horizontal em questão. Você deve guardar cada um destes segmentos de reta em uma lista.

Tipicamente, aplicativos de impressão 3D calculam, a partir destes segmentos de retas como deve ser a fatia a ser impressa e comunicariam-se com a impressora para que a impressão possa ser realizada. Entretanto, esta tarefa é bastante complicada para um projeto desenvolvido no segundo semestre da graduação. Portanto, para simplificar o que seu programa deve fazer é apenas gerar como saída uma imagem no formato .pgm (no formato texto) com o desenho de todos os segmentos de retas encontrados.

### 3 Compilação e entrega do Trabalho

Os trabalhos serão **em dupla** e as apresentações dos trabalhos ocorrerão nos dias 26 e 28 de junho. O dia da apresentação será definido no dia 26 de junho. Todos o programa serão compilados pelo professor com a flag -Wall. A compilação não deve gerar nenhuma *warning*.

Os arquivos .c e .h devem ser enviados para o email (m.v.g.dasilva@gmail.com até o dia 19/06 às 13h00. Será considerado apenas o primeiro envio de email. Portanto, tenha bastante atenção ao enviar o email com o seu trabalho. Você receberá um email de confirmação indicando o recebimento do email.