

## ObjLoader

**Autor: Alex Thomas Almeida Frasson**

**Universidade Federal de Santa Maria – 2012/1**

Após fazer o download de um modelo, recomendo que o importe para um software de edição e exporte para garantir que o mesmo possua todas as normais. Sem as normais a iluminação ficará comprometida. Blender é um bom e pequeno software de modelagem gratuito que pode ser utilizado para esse fim. Também é possível implementar um método para gerar as normais após o carregamento do modelo.

Tenha certeza de que o arquivo mtl e as texturas(se houver) estão com os caminhos relativos ao arquivo obj.

```
1 # Blender v2.63 (sub 0) OBJ File: 'knight_0.blend'
2 # www.blender.org
3 mtlllib knight.mtl
4 v 4.440492 -1.566779 2.318132
5 v 4.476376 -1.564204 3.051705
6 v 1.809036 -1.617785 4.262507
7 v 1.280706 -1.630252 3.752475
8 v 5.208643 2.668229 2.509656
9 v 5.130347 2.186598 3.233362
10 v 1.390684 2.111476 4.930931
11 v 0.778558 2.579238 4.520634
12 v 2.561520 -3.569692 2.919473
13 v 2.887661 -3.336847 3.550022
```

**Arquivo mtl com caminho relativo.**

Devido as informações serem armazenadas na classe `std::vector` o desempenho é menor em DEBUG.

**Ao exportar o modelo de um editor (Blender, 3D Studio Max, Maya...):**

- Exporte as normais para que a iluminação funcione corretamente.
- Escolha a opção para exportar os materiais com caminhos relativos.

**Instruções:**

- Incluir `ObjLoader.h`.
- Crie uma classe para representar um objeto em seu programa e faça com que ela herde da classe `_ObjMesh`. Exemplo: `class MeuObjeto : public _ObjMesh {}`
- Utilize as informações dos vértices, normais e coordenadas de textura para desenhar o modelo.
- Para carregar um arquivo:  
`MeuObjeto obj;`  
`obj.load("C:/Users/Fulano/Desktop/arquivo.obj");`

O vetor `std::vector<_ObjFace> f` armazena as faces do modelo. Cada face possui índices para os vértices(`std::vector<_ObjPoint3f> v`), para as normais(`std::vector<_ObjPoint3f> n`) e

para as coordenadas de textura(`std::vector<_ObjUV> uv`) e também uma referência para um material. Caso a referência para o material não seja nula, pode-se utilizar as informações (cores e texturas) armazenadas pela classe `_ObjMaterial`.

#### **Acessando as informações:**

```
f[0]           //face 0
f[0].v[0]      //índice(posição) do vertice 0 da face 0
v[f[0].v[0]]   //vertice 0 da face 0
n[f[0].n[0]]   //normal do vertice 0 da face 0
```

#### **Percorrendo as faces:**

```
for(int i = 0; i < f.size(); i++)           //percorre todas as faces
{
    for(int j = 0; j < 3; j++)               //percorre todos os vértices da face i
    {
        glNormal(n[f[i].n[j]]);
        glVertex(v[f[i].v[j]]);             //desenha os vértices da face[i]
    }
}
```