

```

obj.h
1 //
2 // コンピュータグラフィックス特論II
3 // 幾何形状データ (Obj形式) の読み込み&描画のサンプルプログラム
4 //
5
6 #ifndef _OBJ_H_
7 #define _OBJ_H_
8
9
10 // ベクトルデータ
11 struct Vector
12 {
13     float x, y, z;
14 };
15
16
17 // カラーデータ
18 struct Color
19 {
20     float r, g, b;
21 };
22
23
24 // 幾何形状データの例 (本プログラムでは使用しない)
25 struct SampleGeometry
26 {
27     int num_vertices; // 頂点数
28     Vector * vertices; // 頂点座標配列 [num_vertices]
29     Vector * normals; // 法線ベクトル配列 [num_vertices]
30     Color * colors; // カラー配列 [num_vertices]
31
32     int num_triangles; // 三角面数
33     int * triangles; // 三角面の頂点番号配列 [num_triangles*3]
34 };
35
36
37 // マテリアルデータ (Mtl形式用)
38 struct Mtl
39 {
40     char * name; // マテリアル名
41     Color kd; // 拡散反射光 (とりあえず拡散反射光を glColor3f() で使用する)
42     char * texture_name; // テクスチャ画像のファイル名
43 };
44
45 // 幾何形状データ (Obj形式用)
46 struct Obj
47 {
48     int num_vertices; // 頂点数
49     Vector * vertices; // 頂点座標配列 [num_vertices]
50
51     int num_normals;
52     Vector * normals; // 法線ベクトル配列 [num_normals]
53
54     int num_tex_coords;
55     Vector * tex_coords; // テクスチャ座標配列 [num_tex_coords]
56
57     int num_triangles; // 三角面数
58     int * tri_v_no; // 三角面の各頂点の頂点座標番号配列 [num_triangles*3]
59     int * tri_vn_no; // 三角面の各頂点の法線ベクトル番号配列 [num_triangles*3]
60     int * tri_vt_no; // 三角面の各頂点のテクスチャ座標番号配列 [num_triangles*3]
61     Mtl ** tri_material; // 三角面の素材 [num_triangles]
62
63     int num_materials; // マテリアル数
64     Mtl ** materials; // マテリアルの配列 [num_materials]
65 };
66
67 // Objファイルの読み込み
68 Obj * LoadObj( const char * filename );
69
70 // Mtlファイルの読み込み
71 void LoadMtl( const char * filename, Obj * obj );
72
73 // オブジェクトのスケールリング (スケールリング後の中心の高さを返す)
74 float ScaleObj( Obj * obj, float max_size );
75
76 // Obj形状データの描画
77 void RenderObj( Obj * obj );
78
79 #endif // _OBJ_H_
80
81
82
83
84
85
86

```