

```

WavefrontObj.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
7 #ifndef _WAVEFRONT_OBJ_H_
8 #define _WAVEFRONT_OBJ_H_
9
10
11 #include <vector>
12 #include <map>
13 #include <string>
14
15 using namespace std;
16
17
18
19 //
20 // Alias|Wavefront Obj形式の幾何形状データ
21 //
22 class WavefrontObj
23 {
24 public:
25     /* 構造体 */
26
27     // 頂点
28     struct Vertex
29     {
30         double x, y, z;
31     };
32
33     // カラー
34     struct Color
35     {
36         double r, g, b;
37     };
38
39     struct Face;
40
41     // グループ
42     struct Group
43     {
44         string name;
45         vector< int > faces;
46     };
47
48     // マテリアル
49     struct Material
50     {
51         string name;
52         Color ambient;
53         Color diffuse;
54         Color specular;
55         string diffuse_map;
56         vector< int > faces;
57     };
58
59     // エレメント
60     struct Element
61     {
62         Group * group;
63         Material * material;
64     };
65
66     // 頂点
67     struct Face : public Element
68     {
69         vector< int > data;
70
71         int NumVertex() const { return data.size() / 3; }
72         int GetVertex( int n ) const { return data[ n*3 ]; }
73         int GetTexture( int n ) const { return data[ n*3+1 ]; }
74         int GetNormal( int n ) const { return data[ n*3+2 ]; }
75     };
76
77
78 private:
79     /* 内部データ */
80     vector< Group * > groups;
81     vector< Material * > materials;
82     vector< Vertex > vertices;
83     vector< Vertex > normals;
84     vector< Vertex > t_coords;
85     vector< Face * > faces;
86
87     map< string, Group * > group_index;
88     map< string, Material * > mtl_index;
89
90
91 public:
92     /* コンストラクタ・デストラクタ */
93     WavefrontObj( const char * file_name );
94     ~WavefrontObj();
95
96 protected:
97     // マテリアルファイルの読み込み
98     void LoadMaterialFile( const char * file_name );
99
100 public:
101     /* アクセサ */
102     const int NumGroups() const { return groups.size(); }
103     const Group * GetGroup( int n ) const { return groups[n]; }
104     const Group * GetGroup( const string & s ) const {
105         return ( (group_index.find( s ) != group_index.end()) ? (*group_index.find( s )).second : NULL ); }
106     const int NumMaterials() const { return materials.size(); }
107     const Material * GetMaterial( int n ) const { return materials[n]; }
108     const int NumVertices() const { return vertices.size(); }
109     const Vertex & GetVertex( int n ) const { return vertices[n]; }
110     const int NumNormals() const { return normals.size(); }
111     const Vertex & GetNormal( int n ) const { return normals[n]; }
112     const int NumTextureCoords() const { return t_coords.size(); }

```

WavefrontObj.h

```
113     const Vertex & GetTextureCoords( int n ) const { return t_coords[n]; }
114     const int NumFaces() const { return faces.size(); }
115     const Face * GetFace( int n ) const { return faces[n]; }
116
117     // OpenGLを使用してオブジェクトを描画
118     void Draw();
119 };
120
121
122
123 #endif // _WAVEFRONT_OBJ_H_
124
```