

grid3D.h

```
1 // #pragma once
2
3
4 #include "Class_Vector.h"
5 #include <vector>
6
7
8 int const maxpasadas=20;
9 extern float factorV, factorVh;
10 extern int MODO_NumerasH, MODO_NumerasFF;
11 extern vector<double> FF, UU, VV, WW;
12
13 class grid3D;
14
15 class R3{
16 public:
17     double L,x,y,z; // L se usa para trazos que unen dos puntos en los poligonos
18
19 //-----
20     void save(ofstream &myfile);
21     void read(ifstream &myfile);
22
23 };
24
25 double ppunto(R3 a, R3 b);
26
27 double ppuntodiff(R3 a, R3 b, R3 c);
28
29 class PoligonoPlano {
30 public:
31     double Area;
32     vector<R3> punto;
33     R3 normal;
34     double Dab;
35     R3 centro;
36
37 //-----
38     void save(ofstream &myfile);
39     void read(ifstream &myfile);
40
41 };
42
43 class Vertex3D {
44 public:
45     double x,y,z;
46     int no;
47     grid3D *papa;
48
49 //----
50     void save(ofstream &myfile);
51     void read(ifstream &myfile, grid3D *papaL);
52 };
53
54 class Cara3D {
55 public:
56     int nv,nh;
57     int iv[4],ih[2];
```

grid3D.h

```

58     int no;
59     double BC;
60     double BC2;
61     int iBC;
62     double Area;
63     grid3D *papa;
64     R3 centro;
65     R3 normal;
66     PoligonoPlano vPolig;
67     int nVertexPolig;
68     double Dij;
69     double nx,ny,nz;
70
71 //----
72     void save(ofstream &myfile);
73     void read(istream &myfile,grid3D *papaL);
74
75     void drawGL();
76 };
77
78
79 class Hexa3D {
80 public:
81     vector<PoligonoPlano> Poligono;
82     vector<int> vecino;
83     vector<int> tipo_vecino;
84     vector<int> dibujado;
85
86     int iv[8];
87     int icara[6];
88     int no;
89     R3 centro;
90     grid3D *papa;
91
92 //-----
93     void save(ofstream &myfile);
94     void read(istream &myfile,grid3D *papaL);
95
96     void draw_caraGL(int ,int,int,int);
97     void draw_caraGL(vector<double> F,double,double,int ,int,int,int);
98     void draw_edgeGL(int i0,int i1);
99 };
100
101
102 class grid3D
103 {
104 public:
105     vector<Vertex3D> v3D;
106     vector<Hexa3D> h3D;
107     vector<Cara3D> Cara;
108     int nH3D,nV3D,nCaras,nPoligonos;
109     double xmin,xmax,ymin,ymax,zmin,zmax;
110     int QuienGeneraPoligonos;
111
112 //-----
113
114     void save(ofstream &myfile);

```

grid3D.h

```

115     void read(ifstream &myfile);
116
117     void minmax();
118
119     void drawGL();
120     void drawGL(vector<double> F);
121     void drawVoronoi();
122
123     void drawVelGL(vector<double>,vector<double>,vector<double>);
124     void drawVelGL2(vector<double>,vector<double>,vector<double>);
125
126     void GeneraCaras(int inicia=false);
127     int AddCara(int ib,int i0,int i1,int i2,int i3);
128     void draw_caraGL(int ii[4]);
129
130     void draw_caraGL(vector<double>F,double minF,double maxF,int ii[4]);
131     void generaPoligonos(int CuantosPoligonos);
132     void generaPoligonos2(int CuantosPoligonos);
133     void CentroCarasBloques();
134
135
136
137     void cubo(int,int,int,float=1,float=1,float=1);
138     void Junta(grid3D g1,grid3D g2);
139     void Junta(grid3D g2);
140     void Rota90Z();
141     void Traslada(double dx,double dy,double dz);
142     grid3D(void);
143     ~grid3D(void);
144
145 };
146 #define ES_BLOQUE 1
147 #define ES_CARA 2
148

```