My Project

Generated by Doxygen 1.8.14

Contents

1 Class Index **1**

1.1 Class List 1

2 File Index **3**

2.1 File List 3

3 Class Documentation **5**

3.1 coordinate Struct Reference 5

3.1.1 Detailed Description 5

3.1.2 Constructor & Destructor Documentation 5

3.1.2.1 coordinate() [1/2] 5

3.1.2.2 coordinate() [2/2] 6

3.1.3 Member Function Documentation 6

3.1.3.1 equals() 6

3.1.3.2 operator==() 6

3.1.4 Member Data Documentation 7

3.1.4.1 x 7

3.1.4.2 y 7

3.1.4.3 z 7

3.2 coordinateHasher Struct Reference 7

3.2.1 Detailed Description 7

3.2.2 Member Function Documentation 8

3.2.2.1 operator()() 8

3.3 coordinatePair Struct Reference 8

3.3.1 Detailed Description . . . 8
3.3.2 Constructor & Destructor Documentation . . . 8
3.3.2.1 coordinatePair() [1/3] . . . 8
3.3.2.2 coordinatePair() [2/3] . . . 9
3.3.2.3 coordinatePair() [3/3] . . . 9
3.3.3 Member Function Documentation . . . 9
3.3.3.1 convertToStandardForm() . . . 9
3.3.3.2 operator==() . . . 10
3.3.4 Member Data Documentation . . . 10
3.3.4.1 v1 . . . 10
3.3.4.2 v2 . . . 10
3.4 edge Struct Reference . . . 10
3.4.1 Detailed Description . . . 11
3.4.2 Constructor & Destructor Documentation . . . 11
3.4.2.1 edge() [1/2] . . . 11
3.4.2.2 edge() [2/2] . . . 11
3.4.3 Member Function Documentation . . . 12
3.4.3.1 convertToStandardForm() . . . 12
3.4.4 Member Data Documentation . . . 12
3.4.4.1 hiddenXY . . . 12
3.4.4.2 hiddenYZ . . . 12
3.4.4.3 hiddenZX . . . 13
3.4.4.4 v1 . . . 13
3.4.4.5 v2 . . . 13
3.5 edges Class Reference . . . 13
3.5.1 Detailed Description . . . 14
3.5.2 Member Function Documentation . . . 14
3.5.2.1 addEdge() [1/2] . . . 14
3.5.2.2 addEdge() [2/2] . . . 14
3.5.2.3 breakEdges() . . . 14

3.5.2.4 changeFrame() . . . 15
3.5.2.5 correctMesh() . . . 15
3.5.2.6 getEdges() . . . 16
3.5.2.7 getProbableEdges() . . . 16
3.5.2.8 getProbableVertices() . . . 16
3.5.2.9 getVertices() . . . 17
3.5.3 Member Data Documentation . . . 17
3.5.3.1 e . . . 17
3.5.3.2 v . . . 17
3.6 face Struct Reference . . . 17
3.6.1 Detailed Description . . . 18
3.6.2 Member Data Documentation . . . 18
3.6.2.1 corrPlane . . . 18
3.6.2.2 edges . . . 18
3.7 faces Class Reference . . . 18
3.7.1 Detailed Description . . . 19
3.7.2 Constructor & Destructor Documentation . . . 19
3.7.2.1 faces() . . . 19
3.7.3 Member Function Documentation . . . 20
3.7.3.1 addFace() . . . 20
3.7.3.2 changeFrame() . . . 20
3.7.3.3 checkHidden() . . . 20
3.7.3.4 correctFaceOrientation() . . . 21
3.7.3.5 getEdges() . . . 21
3.7.3.6 getFaces() . . . 21
3.7.3.7 makeFaces() . . . 22
3.7.3.8 removePseudoFaces() . . . 22
3.7.4 Member Data Documentation . . . 22
3.7.4.1 e . . . 23
3.7.4.2 f . . . 23

3.8 plane Struct Reference . . . 23
3.8.1 Detailed Description . . . 23
3.8.2 Member Data Documentation . . . 23
3.8.2.1 a . . . 24
3.8.2.2 b . . . 24
3.8.2.3 c . . . 24
3.8.2.4 d . . . 24
3.9 vertices Class Reference . . . 24
3.9.1 Detailed Description . . . 25
3.9.2 Member Function Documentation . . . 25
3.9.2.1 addVertex() [1/3] . . . 25
3.9.2.2 addVertex() [2/3] . . . 25
3.9.2.3 addVertex() [3/3] . . . 26
3.9.2.4 changeFrame() . . . 26
3.9.2.5 getVertices() . . . 26
3.9.3 Member Data Documentation . . . 26
3.9.3.1 v . . . 27

4 File Documentation **29**

4.1 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/edges.h File Reference . . . 29
4.1.1 Macro Definition Documentation . . . 29
4.1.1.1 EDGES_H . . . 29
4.2 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/faces.h File Reference . . . 29
4.3 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/inout.h File Reference . . . 30
4.3.1 Function Documentation . . . 30
4.3.1.1 input() . . . 30
4.3.1.2 output() . . . 30
4.4 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/objects.h File Reference . . . 31
4.4.1 Function Documentation . . . 32

4.4.1.1 operator() . . . 32
4.4.1.2 operator()() . . . 32
4.4.1.3 vertex() [1/3] . . . 32
4.4.1.4 vertex() [2/3] . . . 32
4.4.1.5 vertex() [3/3] . . . 33
4.4.2 Variable Documentation . . . 33
4.4.2.1 frame_old . . . 33
4.4.2.2 numEdges . . . 33
4.4.2.3 v . . . 33
4.5 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/vertices.h File Reference . . . 34
4.5.1 Variable Documentation . . . 34
4.5.1.1 operator . . . 34
4.6 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/edge.cpp File Reference . . . 34
4.7 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/edges.cpp File Reference . . . 34
4.7.1 Function Documentation . . . 35
4.7.1.1 compareFloats() . . . 35
4.8 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/face.cpp File Reference . . . 35
4.9 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/faces.cpp File Reference . . . 35
4.10 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/input.cpp File Reference . . . 35
4.10.1 Function Documentation . . . 35
4.10.1.1 input() . . . 35
4.11 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/main.cpp File Reference . . . 36
4.11.1 Function Documentation . . . 36
4.11.1.1 main() . . . 36
4.12 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/output.cpp File Reference . . . 36
4.12.1 Function Documentation . . . 37
4.12.1.1 output() . . . 37
4.13 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/vertex.cpp File Reference . . . 37
4.14 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/vertices.cpp File Reference . . . 37

Index **39**

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

coordinate
: Struct representing only the coordinates of a point 5

coordinateHasher
: Struct containing Hash function for coodinate type 7

coordinatePair
: Struct representing only the coordinates of end points of edge 8

edge
: Struct representing directed edge from v1 to v2 10

edges
: Class representing list of edges 13

face
: Struct representing polygon face 17

faces
: Class representing list of faces of 3D object 18

plane
: Struct representing plane of form ax+by+cz=d 23

vertices
: Class representing list of vertices 24

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/edges.h . . . 29
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/faces.h 29
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/inout.h 30
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/objects.h . . . 31
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/vertices.h . . 34
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/edge.cpp 34
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/edges.cpp 34
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/face.cpp 35
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/faces.cpp 35
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/input.cpp 35
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/main.cpp 36
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/output.cpp 36
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/vertex.cpp 37
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/vertices.cpp . . . 37

Chapter 3

Class Documentation

3.1 coordinate Struct Reference

Struct representing only the coordinates of a point.

`#include <objects.h>`

Public Member Functions

- coordinate ()
  Constructor to initialize object of coordinate.
- coordinate (float, float, float)
  Constructor to initialize object of coordinate.
- bool equals (coordinate)
  Function to check equality of 2 objects of coordinate type.
- bool operator== (const coordinate &other) const
  Checks for equality of 2 coordinate type objects.

Public Attributes

- float x
- float y
- float z

3.1.1 Detailed Description

Struct representing only the coordinates of a point.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 coordinate() [1/2]

`coordinate::coordinate ( )`

Constructor to initialize object of coordinate.

Parameters

void	

3.1.2.2 coordinate() [2/2]

```
coordinate::coordinate (
            float x,
            float y,
            float z )
```

Constructor to initialize object of coordinate.

Parameters

coordinates	x, y, z

3.1.3 Member Function Documentation

3.1.3.1 equals()

```
bool coordinate::equals (
            coordinate  )
```

Function to check equality of 2 objects of coordinate type.

Parameters

coordinates	of the end points

3.1.3.2 operator==()

```
bool coordinate::operator== (
            const coordinate & other ) const  [inline]
```

Checks for equality of 2 coordinate type objects.

Parameters

Coordinate	to be compared with the current coordinate

Returns

boolean that returns true if the 2 objects are equal

3.1.4 Member Data Documentation

3.1.4.1 x

```
float coordinate::x
```

3.1.4.2 y

```
float coordinate::y
```

3.1.4.3 z

```
float coordinate::z
```

The documentation for this struct was generated from the following files:

- /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/objects.h
- /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/vertex.cpp

3.2 coordinateHasher Struct Reference

Struct containing Hash function for coodinate type.

```
#include <objects.h>
```

Public Member Functions

- std::size_t operator() (const coordinate &k) const

3.2.1 Detailed Description

Struct containing Hash function for coodinate type.

3.2.2 Member Function Documentation

3.2.2.1 operator()()

```
std::size_t coordinateHasher::operator() (
            const coordinate & k ) const  [inline]
```

The documentation for this struct was generated from the following file:

- /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/objects.h

3.3 coordinatePair Struct Reference

Struct representing only the coordinates of end points of edge.

`#include <objects.h>`

Public Member Functions

- coordinatePair ()
  Constructor to initialize object of coordinatePair.
- coordinatePair (coordinate, coordinate)
  Constructor to initialize object of coordinatePair.
- coordinatePair (float, float, float, float, float, float)
- void convertToStandardForm ()
  Converts pair of vertices in standard form.
- bool operator== (const coordinatePair &other) const
  Checks for equality of 2 coordinatePair type objects.

Public Attributes

- coordinate v1
- coordinate v2

3.3.1 Detailed Description

Struct representing only the coordinates of end points of edge.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 coordinatePair() [1/3]

```
coordinatePair::coordinatePair ( )
```

Constructor to initialize object of coordinatePair.

Parameters

void	

3.3.2.2 coordinatePair() [2/3]

```
coordinatePair::coordinatePair (
            coordinate v1,
            coordinate v2 )
```

Constructor to initialize object of coordinatePair.

Parameters

v1	and v2 are coordinates of end points

3.3.2.3 coordinatePair() [3/3]

```
coordinatePair::coordinatePair (
            float ,
            float ,
            float ,
            float ,
            float ,
            float  )
```

3.3.3 Member Function Documentation

3.3.3.1 convertToStandardForm()

```
void coordinatePair::convertToStandardForm ( )
```

Converts pair of vertices in standard form.

Parameters

a	is the coordinate to be added, numEdges is the number of edges that the vertex is part of

Returns

boolean that returns true if face is valid and not already present

1st vertex should have lower value of coordinates than the 2nd vertex (priority order -> x>y>z)

3.3.3.2 operator==()

```
bool coordinatePair::operator== (
            const coordinatePair & other ) const  [inline]
```

Checks for equality of 2 coordinatePair type objects.

Parameters

coordinatePair	to be compared with the current coordinatePair

Returns

boolean that returns true if the 2 objects are equal

3.3.4 Member Data Documentation

3.3.4.1 v1

```
coordinate coordinatePair::v1
```

3.3.4.2 v2

```
coordinate coordinatePair::v2
```

The documentation for this struct was generated from the following files:

- /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/objects.h
- /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/edge.cpp

3.4 edge Struct Reference

Struct representing directed edge from v1 to v2.

```
#include <objects.h>
```

Public Member Functions

- edge (coordinate v1, coordinate v2)

  Constructor to initialize object of edge.
- edge (float, float, float, float, float, float)
- void convertToStandardForm ()

  Converts edge in standard form.

Public Attributes

- coordinate v1
- coordinate v2
- bool hiddenXY

  True if edge is hidden in XY projection.
- bool hiddenYZ

  True if edge is hidden in XY projection.
- bool hiddenZX

  True if edge is hidden in XY projection.

3.4.1 Detailed Description

Struct representing directed edge from v1 to v2.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 edge() [1/2]

```
edge::edge (
            coordinate v1,
            coordinate v2 )
```

Constructor to initialize object of edge.

Parameters

coordinates	of end points

3.4.2.2 edge() [2/2]

```
edge::edge (
            float ,
```

```
            float ,
            float ,
            float ,
            float ,
            float  )
```

3.4.3 Member Function Documentation

3.4.3.1 convertToStandardForm()

```
void edge::convertToStandardForm ( )
```

Converts edge in standard form.

Parameters

a	is the coordinate to be added, numEdges is the number of edges that the vertex is part of

Returns

boolean that returns true if face is valid and not already present

1st vertex should have lower value of coordinates than the 2nd vertex (priority order -> x>y>z)

3.4.4 Member Data Documentation

3.4.4.1 hiddenXY

```
bool edge::hiddenXY
```

True if edge is hidden in XY projection.

3.4.4.2 hiddenYZ

```
bool edge::hiddenYZ
```

True if edge is hidden in XY projection.

3.4.4.3 hiddenZX

```
bool edge::hiddenZX
```

True if edge is hidden in XY projection.

3.4.4.4 v1

```
coordinate edge::v1
```

3.4.4.5 v2

```
coordinate edge::v2
```

The documentation for this struct was generated from the following files:

- /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/objects.h
- /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/edge.cpp

3.5 edges Class Reference

Class representing list of edges.

```
#include <edges.h>
```

Public Member Functions

- bool addEdge (edge)

  Adds new edge to the list of edges.
- bool addEdge (int, int, int, int, int, int)
- void changeFrame (plane, plane, plane)

  Transform all edges according to new coordinate system.
- void breakEdges (bool, int)

  Break edges for preprocessing.
- void getProbableEdges (edges *, edges *, edges *)

  Gives all probable edges.
- void getProbableVertices (vertices *, vertices *, vertices *)

  Gives all probable vertices.
- void correctMesh ()

  Corrects the mesh to remove probable edges.
- vertices & getVertices ()

  Returns the list of vertices.
- unordered_map< coordinatePair, edge, coordinatePairHasher > & getEdges ()

  Returns the list of edges.

Private Attributes

- unordered_map< coordinatePair, edge, coordinatePairHasher > e

  Hashmap of list of edges.
- vertices v

  list of vertices

3.5.1 Detailed Description

Class representing list of edges.

3.5.2 Member Function Documentation

3.5.2.1 addEdge() [1/2]

```
bool edges::addEdge (
            edge a )
```

Adds new edge to the list of edges.

Parameters

a	is the edge to be added

Returns

boolean that returns true if edge is valid and not already present

3.5.2.2 addEdge() [2/2]

```
bool edges::addEdge (
            int ,
            int ,
            int ,
            int ,
            int ,
            int  )
```

3.5.2.3 breakEdges()

```
void edges::breakEdges (
            bool flag,
            int view )
```

Break edges for preprocessing.

Parameters

flag	is true for 2D to 3D and false for 3D to 2D

Returns

void

It traverses for all pairs of edges and looks for intersection in the projections and adds new edges to the list e and new vertex to v. It retains the original edges if flag is true (2D to 3D) but discards the original edges if flag is false (3D to 2D).

3.5.2.4 changeFrame()

```
void edges::changeFrame (
            plane a,
            plane b,
            plane c )
```

Transform all edges according to new coordinate system.

Parameters

a,b	and c are the 3 projection planes that are to be treated as XY, YZ and ZX planes respectively

Returns

void

3.5.2.5 correctMesh()

```
void edges::correctMesh ( )
```

Corrects the mesh to remove probable edges.

Parameters

void	

Returns

void

It traverses across the set of edges e and removes the edges that have any of the endpoints having numEdges $<$ 3.

3.5.2.6 getEdges()

```
unordered_map< coordinatePair, edge, coordinatePairHasher > & edges::getEdges ( )
```

Returns the list of edges.

Parameters

void	

Returns

Reference to the list of edges e

3.5.2.7 getProbableEdges()

```
void edges::getProbableEdges (
            edges * et,
            edges * ef,
            edges * es )
```

Gives all probable edges.

Parameters

void	

Returns

void

It generates all edges that are probable from the given projections (2D to 3D)and adds them to e.

3.5.2.8 getProbableVertices()

```
void edges::getProbableVertices (
            vertices * vt,
            vertices * vf,
            vertices * vs )
```

Gives all probable vertices.

Parameters

void	

Returns

void

It generates all vertices that are probable from the given projections (2D to 3D)and adds them to v.

3.5.2.9 getVertices()

```
vertices & edges::getVertices ( )
```

Returns the list of vertices.

Parameters

void	

Returns

Reference to the list of vertices

3.5.3 Member Data Documentation

3.5.3.1 e

```
unordered_map<coordinatePair,edge,coordinatePairHasher> edges::e  [private]
```

Hashmap of list of edges.

3.5.3.2 v

```
vertices edges::v  [private]
```

list of vertices

The documentation for this class was generated from the following files:

- /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/edges.h
- /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/edges.cpp

3.6 face Struct Reference

Struct representing polygon face.

```
#include <objects.h>
```

Public Attributes

- vector< pair(coordinate, ccordinate)> edges

  List of edges constituting the face.
- plane corrPlane

  Corresponding infinte plane of face.

3.6.1 Detailed Description

Struct representing polygon face.

3.6.2 Member Data Documentation

3.6.2.1 corrPlane

```
plane face::corrPlane
```

Corresponding infinte plane of face.

3.6.2.2 edges

```
vector<pair(coordinate,ccordinate)> face::edges
```

List of edges constituting the face.

The documentation for this struct was generated from the following file:

- /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/objects.h

3.7 faces Class Reference

Class representing list of faces of 3D object.

```
#include <faces.h>
```

Public Member Functions

- faces (edges *)
  Constructor to initialize object of faces.
- void addFace (face *)
  Adds a face to the list of faces f.
- void changeFrame (plane, plane, plane)
  Transform all faces according to new coordinate system.
- void checkHidden (edge *)
  Checks if given edge is hidden.
- void makeFaces ()
  Makes all possible faces and adds to f.
- void removePseudoFaces ()
  Removes Pseudo faces.
- void correctFaceOrientation ()
  Corrects orientation of each face.
- edges & getEdges ()
  Returns the list of edges.
- vector< face > & getFaces ()
  Returns the list of faces.

Private Attributes

- vector< face > f
  list of faces
- edges e

3.7.1 Detailed Description

Class representing list of faces of 3D object.

3.7.2 Constructor & Destructor Documentation

3.7.2.1 faces()

```
faces::faces (
            edges * e )
```

Constructor to initialize object of faces.

Parameters

e	is the set of edges

Returns

void

3.7.3 Member Function Documentation

3.7.3.1 addFace()

```
bool faces::addFace (
            face * a )
```

Adds a face to the list of faces f.

Parameters

a	is the face to be added

Returns

boolean that returns true if face is valid and not already present

3.7.3.2 changeFrame()

```
void faces::changeFrame (
            plane a,
            plane b,
            plane c )
```

Transform all faces according to new coordinate system.

Parameters

a,b	and c are the 3 projection planes that are to be treated as XY, YZ and ZX planes respectively

Returns

void

3.7.3.3 checkHidden()

```
void faces::checkHidden (
            edge * a )
```

Checks if given edge is hidden.

Parameters

void	

Returns

void

It compares edge a with all faces to check if it is hidden by any of the faces

3.7.3.4 correctFaceOrientation()

```
void faces::correctFaceOrientation ( )
```

Corrects orientation of each face.

Parameters

void	

Returns

void

It finds the correct orientation of each face of the solid using Moebius Rule

3.7.3.5 getEdges()

```
edges & faces::getEdges ( )
```

Returns the list of edges.

Parameters

void	

Returns

Reference to the list of edges

3.7.3.6 getFaces()

```
vector< face > & faces::getFaces ( )
```

Returns the list of faces.

Parameters

void	

Returns

Reference to the list of faces f

3.7.3.7 makeFaces()

```
void faces::makeFaces ( )
```

Makes all possible faces and adds to f.

Parameters

void	

Returns

void

It makes all possible faces using Minimum Internal Angle algorithm and adds the faces to f.

3.7.3.8 removePseudoFaces()

```
void faces::removePseudoFaces ( )
```

Removes Pseudo faces.

Parameters

void	

Returns

void

It removes all the Pseudo faces using Decision Chaining algorithm

3.7.4 Member Data Documentation

3.7.4.1 e

```
edges faces::e  [private]
```

3.7.4.2 f

```
vector<face> faces::f  [private]
```

list of faces

The documentation for this class was generated from the following files:

- /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/faces.h
- /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/faces.cpp

3.8 plane Struct Reference

Struct representing plane of form ax+by+cz=d.

```
#include <objects.h>
```

Public Attributes

- float a
  a in ax+by+cz=d
- float b
  b in ax+by+cz=d
- float c
  c in ax+by+cz=d
- float d
  d in ax+by+cz=d

3.8.1 Detailed Description

Struct representing plane of form ax+by+cz=d.

3.8.2 Member Data Documentation

3.8.2.1 a

`float plane::a`

a in ax+by+cz=d

3.8.2.2 b

`float plane::b`

b in ax+by+cz=d

3.8.2.3 c

`float plane::c`

c in ax+by+cz=d

3.8.2.4 d

`float plane::d`

d in ax+by+cz=d

The documentation for this struct was generated from the following file:

- /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/objects.h

3.9 vertices Class Reference

Class representing list of vertices.

`#include <vertices.h>`

Public Member Functions

- void changeFrame (plane, plane, plane)

  Transform all vertices according to new coordinate system.
- bool addVertex (vertex)

  Adds vertex to the list of vertices.
- bool addVertex (coordinate, int)

  Adds vertex to the list of vertices.
- bool addVertex (int, int, int, int)
- unordered_map< coordinate, vertex, coordinateHasher > & getVertices ()

  Returns the list of vertices.

Private Attributes

- unordered_map< coordinate, vertex, coordinateHasher > v
  list of vertices

3.9.1 Detailed Description

Class representing list of vertices.

3.9.2 Member Function Documentation

3.9.2.1 addVertex() [1/3]

```
bool vertices::addVertex (
            vertex a )
```

Adds vertex to the list of vertices.

Parameters

a	is the vertex to be added

Returns

boolean that returns true if face is valid and not already present

3.9.2.2 addVertex() [2/3]

```
bool vertices::addVertex (
            coordinate a,
            int numEdges = 0 )
```

Adds vertex to the list of vertices.

Parameters

a	is the coordinate to be added, numEdges is the number of edges that the vertex is part of

Returns

boolean that returns true if face is valid and not already present

3.9.2.3 addVertex() [3/3]

```
bool vertices::addVertex (
            int ,
            int ,
            int ,
            int  )
```

3.9.2.4 changeFrame()

```
void vertices::changeFrame (
            plane a,
            plane b,
            plane c )
```

Transform all vertices according to new coordinate system.

Parameters

a,b	and c are the 3 projection planes that are to be treated as XY, YZ and ZX planes respectively

Returns

void

3.9.2.5 getVertices()

```
unordered_map< coordinate, vertex, coordinateHasher > & vertices::getVertices ( )
```

Returns the list of vertices.

Parameters

void	

Returns

Reference to the list of vertices

3.9.3 Member Data Documentation

3.9.3.1 v

```
unordered_map<coordinate,vertex,coordinateHasher> vertices::v  [private]
```

list of vertices

The documentation for this class was generated from the following files:

- /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/vertices.h
- /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/vertices.cpp

Chapter 4

File Documentation

4.1 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/edges.h File Reference

```
#include "objects.h"
```

Classes

- class edges

  Class representing list of edges.

Macros

- #define EDGES_H

4.1.1 Macro Definition Documentation

4.1.1.1 EDGES_H

```
#define EDGES_H
```

4.2 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/faces.h File Reference

```
#include "objects.h"
#include "edges.h"
```

Classes

- class faces

  Class representing list of faces of 3D object.

4.3 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/inout.h File Reference

```
#include "vertices.h"
#include "faces.h"
#include "edges.h"
```

Functions

- void input (edges *, faces *, vertices *)

  Takes input.
- void output (edges *, faces *, vertices *)

  Gives output.

4.3.1 Function Documentation

4.3.1.1 input()

```
void input (
            edges * e,
            faces * f,
            vertices * v )
```

Takes input.

Parameters

e	is the list of edges, f is the list of faces (NULL for 2D to 3D), v is the list of vertices

Returns

void

4.3.1.2 output()

```
void output (
            edges * ,
```

```
            faces * ,
            vertices *  )
```

Gives output.

Parameters

e	is the list of edges, f is the list of faces (NULL for 3D to 2D), v is the list of vertices (NULL for 3D to 2D)

Returns

void

4.4 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/objects.h File Reference

```
#include <bits/stdc++.h>
```

Classes

- struct plane

  Struct representing plane of form ax+by+cz=d.
- struct coordinate

  Struct representing only the coordinates of a point.
- struct coordinatePair

  Struct representing only the coordinates of end points of edge.
- struct edge

  Struct representing directed edge from v1 to v2.
- struct face

  Struct representing polygon face.
- struct coordinateHasher

  Struct containing Hash function for coodinate type.

Functions

- vertex ()

  Constructor to initialize object of vertex.
- vertex (coordinate, int)

  Constructor to initialize object of vertex.
- vertex (int, int, int, int)

  Constructor to initialize object of vertex.
- struct coordinateHasher operator ()(const coordinatePair &k) const

  Struct containing Hash function for coodinatePair type.
- std::size_t operator() (const coordinate &k) const

Variables

- struct coordinatePair v

  Struct representing vertex (x,y,z)
- float numEdges

  Number of edges adjacent to vertex.
- bool frame_old

  Tells if frame of vertex is new or old.

4.4.1 Function Documentation

4.4.1.1 operator()

```
struct coordinateHasher operator ( ) const &
```

Struct containing Hash function for coodinatePair type.

4.4.1.2 operator()()

```
std::size_t operator::operator() (
            const coordinate & k ) const
```

4.4.1.3 vertex() [1/3]

```
vertex::vertex ( )
```

Constructor to initialize object of vertex.

Parameters

void	

4.4.1.4 vertex() [2/3]

```
vertex::vertex (
            coordinate a,
            int numEdges = 0 )
```

Constructor to initialize object of vertex.

Parameters

coordinate	and number of edges

4.4.1.5 vertex() [3/3]

```
vertex::vertex (
            int x,
            int y,
            int z,
            int numEdges = 0 )
```

Constructor to initialize object of vertex.

Parameters

coordinates	x, y, z and number of neighbours of vertex

4.4.2 Variable Documentation

4.4.2.1 frame_old

```
bool frame_old
```

Tells if frame of vertex is new or old.

4.4.2.2 numEdges

```
float numEdges
```

Number of edges adjacent to vertex.

4.4.2.3 v

```
struct coordinatePair v
```

Struct representing vertex (x,y,z)

4.5 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/vertices.h File Reference

```
#include "objects.h"
```

Classes

- class vertices

  Class representing list of vertices.

Variables

- class vertices operator

4.5.1 Variable Documentation

4.5.1.1 operator

```
class vertices operator
```

4.6 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/edge.cpp File Reference

```
#include "objects.h"
```

4.7 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/edges.cpp File Reference

```
#include "edges.h"
#include "objects.h"
```

Functions

- bool compareFloats (float A, float B)

4.7.1 Function Documentation

4.7.1.1 compareFloats()

```
bool compareFloats (
            float A,
            float B )
```

4.8 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/face.cpp File Reference

4.9 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/faces.cpp File Reference

```
#include "faces.h"
#include "edges.h"
#include "objects.h"
```

4.10 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/input.cpp File Reference

```
#include "objects.h"
#include "vertices.h"
#include "faces.h"
#include "edges.h"
```

Functions

- void input (edges *e, faces *f, vertices *v)

  Takes input.

4.10.1 Function Documentation

4.10.1.1 input()

```
void input (
            edges * e,
            faces * f,
            vertices * v )
```

Takes input.

Parameters

e	is the list of edges, f is the list of faces (NULL for 2D to 3D), v is the list of vertices

Returns

void

4.11 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/main.cpp File Reference

```
#include "faces.h"
#include "edges.h"
#include "vertices.h"
#include "inout.h"
#include "objects.h"
#include <iostream>
#include <bits/stdc++.h>
#include <gtk/gtk.h>
```

Functions

- int main (int argc, char **argv)

4.11.1 Function Documentation

4.11.1.1 main()

```
int main (
            int argc,
            char ** argv )
```

We will use gtk library for graphic input/ouput.

4.12 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/output.cpp File Reference

```
#include "objects.h"
#include "vertices.h"
#include "faces.h"
#include "edges.h"
```

Functions

- void output (edges *, faces *, vertices *)

  Gives output.

4.12.1 Function Documentation

4.12.1.1 output()

```
void output (
            edges * ,
            faces * ,
            vertices *  )
```

Gives output.

Parameters

e	is the list of edges, f is the list of faces (NULL for 3D to 2D), v is the list of vertices (NULL for 3D to 2D)

Returns

void

4.13 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/vertex.cpp File Reference

```
#include "objects.h"
```

4.14 /Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/vertices.cpp File Reference

```
#include "vertices.h"
#include "objects.h"
```

Index

/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/edges.h, 29
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/faces.h, 29
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/inout.h, 30
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/objects.h, 31
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/include/vertices.h, 34
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/edge.cpp, 34
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/edges.cpp, 34
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/face.cpp, 35
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/faces.cpp, 35
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/input.cpp, 35
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/main.cpp, 36
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/output.cpp, 36
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/vertex.cpp, 37
/Users/madhur/Documents/Sem 6/COP290/Assignment 1/git/assignment_1/Code/src/vertices.cpp, 37

a
    plane, 23
addEdge
    edges, 14
addFace
    faces, 20
addVertex
    vertices, 25

b
    plane, 24
breakEdges
    edges, 14

c
    plane, 24
changeFrame
    edges, 15
    faces, 20
    vertices, 26
checkHidden
    faces, 20
compareFloats
    edges.cpp, 35
convertToStandardForm
    coordinatePair, 9
    edge, 12
coordinate, 5
    coordinate, 5, 6
    equals, 6
    operator==, 6
    x, 7
    y, 7
    z, 7
coordinateHasher, 7
    operator(), 8
coordinatePair, 8
    convertToStandardForm, 9
    coordinatePair, 8, 9
    operator==, 10
    v1, 10
    v2, 10
corrPlane
    face, 18
correctFaceOrientation
    faces, 21
correctMesh
    edges, 15

d
    plane, 24

e
    edges, 17
    faces, 22
EDGES_H
    edges.h, 29
edge, 10
    convertToStandardForm, 12
    edge, 11
    hiddenXY, 12
    hiddenYZ, 12
    hiddenZX, 12
    v1, 13
    v2, 13
edges, 13
    addEdge, 14
    breakEdges, 14
    changeFrame, 15

correctMesh, 15
e, 17
face, 18
getEdges, 15
getProbableEdges, 16
getProbableVertices, 16
getVertices, 17
v, 17
edges.cpp
compareFloats, 35
edges.h
EDGES_H, 29
equals
coordinate, 6

f
faces, 23
face, 17
corrPlane, 18
edges, 18
faces, 18
addFace, 20
changeFrame, 20
checkHidden, 20
correctFaceOrientation, 21
e, 22
f, 23
faces, 19
getEdges, 21
getFaces, 21
makeFaces, 22
removePseudoFaces, 22
frame_old
objects.h, 33

getEdges
edges, 15
faces, 21
getFaces
faces, 21
getProbableEdges
edges, 16
getProbableVertices
edges, 16
getVertices
edges, 17
vertices, 26

hiddenXY
edge, 12
hiddenYZ
edge, 12
hiddenZX
edge, 12

inout.h
input, 30
output, 30
input
inout.h, 30
input.cpp, 35
input.cpp
input, 35

main
main.cpp, 36
main.cpp
main, 36
makeFaces
faces, 22

numEdges
objects.h, 33

objects.h
frame_old, 33
numEdges, 33
operator, 32
operator(), 32
v, 33
vertex, 32, 33
operator
objects.h, 32
vertices.h, 34
operator()
coordinateHasher, 8
objects.h, 32
operator==
coordinate, 6
coordinatePair, 10
output
inout.h, 30
output.cpp, 37
output.cpp
output, 37

plane, 23
a, 23
b, 24
c, 24
d, 24

removePseudoFaces
faces, 22

v
edges, 17
objects.h, 33
vertices, 26
v1
coordinatePair, 10
edge, 13
v2
coordinatePair, 10
edge, 13
vertex
objects.h, 32, 33
vertices, 24
addVertex, 25

changeFrame, 26
getVertices, 26
v, 26
vertices.h
operator, 34

x
coordinate, 7

y
coordinate, 7

z
coordinate, 7