My Project

Generated by Doxygen 1.8.13

Contents

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

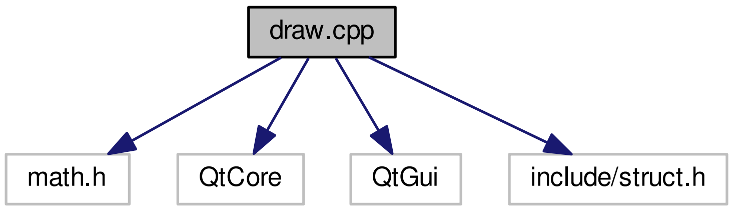
Chapter 2

File Documentation

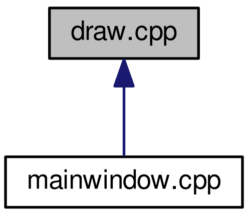
2.1 draw.cpp File Reference

```
#include <math.h>
#include <QtCore>
#include <QtGui>
#include "include/struct.h"
```

Include dependency graph for draw.cpp:



This graph shows which files directly or indirectly include this file:



Functions

- threeView get_three_views (string filename)
- void print_nodes (vector< node > v)
- void print_edges (vector< edge > v)
- coordinate get_average (vector< node > v)
- int get_max_y (vector< edge > v)
- int get_max_x (vector< edge > v)
- int get_min_x (vector< edge > v)
- int get_min_y (vector< edge > v)
- int get_min_z (vector< edge > v)
- int get_max_z (vector< edge > v)
- QPicture draw_views (vector< edge > v)
- QPicture draw_xy (vector< edge > v)
- QPicture draw_yz (vector< edge > v)
- QPicture draw_xz (vector< edge > v)

2.1.1 Function Documentation

2.1.1.1 draw_views()

```
QPicture draw_views (
            vector< edge > v )
```

Takes a graph as an input. Renders view from the same graph.

2.1.1.2 draw_xy()

```
QPicture draw_xy (
            vector< edge > v )
```

Takes a graph description (vector of nodes) as and input. Draws the XY coordinates of this graph.

2.1.1.3 draw_xz()

```
QPicture draw_xz (
            vector< edge > v )
```

Takes a graph description (vector of nodes) as and input. Draws the XZ coordinates of this graph.

2.1.1.4 draw_yz()

```
QPicture draw_yz (
            vector< edge > v )
```

Takes a graph description (vector of nodes) as and input. Draws the YZ coordinates of this graph.

2.1.1.5 get_average()

```
coordinate get_average (
            vector< node > v )
```

Returns a coordinate average of the coordinates of all the nodes in a vector of nodes.

2.1.1.6 get_max_x()

```
int get_max_x (
            vector< edge > v )
```

Helper function that gets maximum x coordinate of a node in a vector of edges.

2.1.1.7 get_max_y()

```
int get_max_y (
            vector< edge > v )
```

Helper function that gets maximum y coordinate of a node in a vector of edges.

2.1.1.8 get_max_z()

```
int get_max_z (
            vector< edge > v )
```

Helper function that gets maximum z coordinate of a node in a vector of edges.

2.1.1.9 get_min_x()

```
int get_min_x (
            vector< edge > v )
```

Helper function that gets minimum x coordinate of a node in a vector of edges.

2.1.1.10 get_min_y()

```
int get_min_y (
            vector< edge > v )
```

Helper function that gets minimum y coordinate of a node in a vector of edges.

2.1.1.11 get_min_z()

```
int get_min_z (
            vector< edge > v )
```

Helper function that gets minimum z coordinate of a node in a vector of edges.

2.1.1.12 get_three_views()

```
threeView get_three_views (
             string filename )
```

Reads the 2D input from a file given as an argument. Converts the input into a graph and returns the graph of three orthographic views

2.1.1.13 print_edges()

```
void print_edges (
             vector< edge > v )
```

Helper function that prints coordinates of two nodes that make up an edge in a vector of edges.

2.1.1.14 print_nodes()

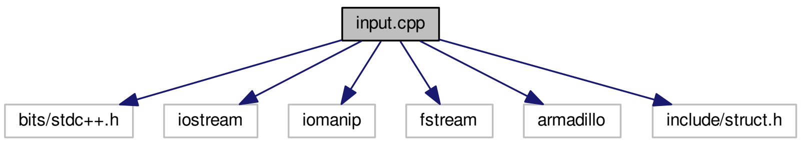
```
void print_nodes (
             vector< node > v )
```

Helper function that prints coordinates of a node in a vector of nodes.

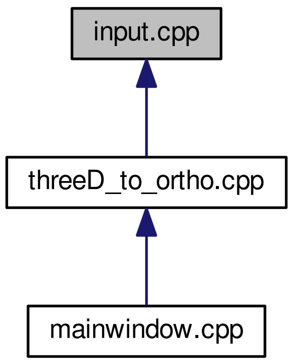
2.2 input.cpp File Reference

```
#include <bits/stdc++.h>
#include <iostream>
#include <iomanip>
#include <fstream>
#include <armadillo>
#include "include/struct.h"
```

Include dependency graph for input.cpp:



This graph shows which files directly or indirectly include this file:



Functions

- graph get_3D_graph (string filename="input_3D.txt")
- mat get_mx4_matrix (vector$<$ node $>$ v, int cols=4)

2.2.1 Function Documentation

2.2.1.1 get_3D_graph()

```
graph get_3D_graph (
            string filename = "input_3D.txt" )
```

Reads the 3D input from a file given as an argument. Converts the input into a graph and returns the graph.

2.2.1.2 get_mx4_matrix()

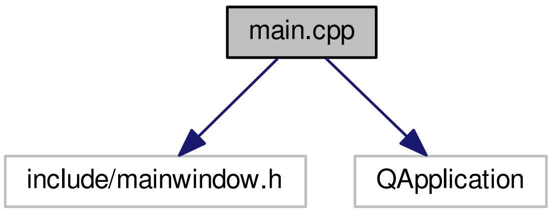
```
mat get_mx4_matrix (
            vector< node > v,
            int cols = 4 )
```

Takes a graph as an input. Converts the graph into a 4X4 coordinate matrix as specified in the mathematical model.

2.3 main.cpp File Reference

```
#include "include/mainwindow.h"
#include <QApplication>
```

Include dependency graph for main.cpp:



Functions

- int main (int argc, char ∗argv[ ])

2.3.1 Function Documentation

2.3.1.1 main()

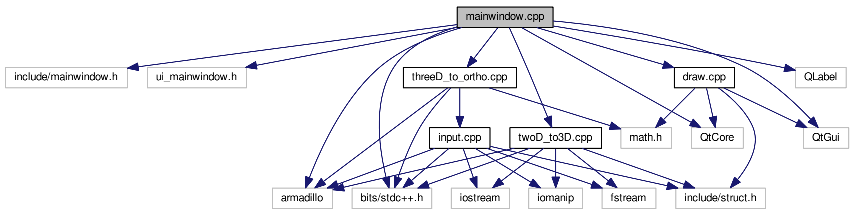
```
int main (
            int argc,
            char ∗ argv[ ] )
```

2.4 mainwindow.cpp File Reference

```
#include "include/mainwindow.h"
#include <ui_mainwindow.h>
#include <bits/stdc++.h>
#include <QtCore>
#include <QtGui>
#include <QLabel>
#include <armadillo>
#include "draw.cpp"
#include "threeD_to_ortho.cpp"
```

```
#include "twoD_to3D.cpp"
```

Include dependency graph for mainwindow.cpp:



Functions

- coordinate add_coord (coordinate c1, coordinate c2)
- void translate_graph (vector< node > &v, coordinate c)

Variables

- vector< node > node_0
- vector< node > node_1
- vector< pair_ > edge_codes_1
- int mode

2.4.1 Function Documentation

2.4.1.1 add_coord()

```
coordinate add_coord (
            coordinate c1,
            coordinate c2 )
```

Helper func that adds two coordinates

2.4.1.2 translate_graph()

```
void translate_graph (
            vector< node > & v,
            coordinate c )
```

Translates graph taking a vector of nodes and an average coordinate

2.4.2 Variable Documentation

2.4.2.1 edge_codes_1

```
vector<pair_> edge_codes_1
```

2.4.2.2 mode

```
int mode
```

2.4.2.3 node_0

```
vector<node> node_0
```

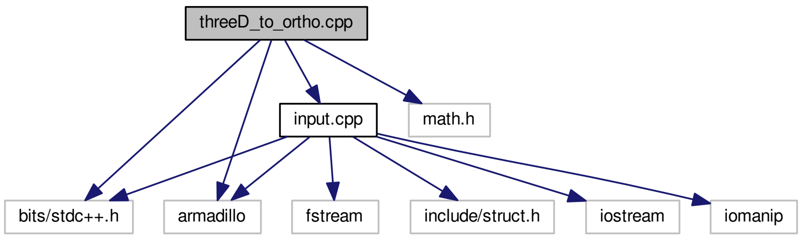
2.4.2.4 node_1

```
vector<node> node_1
```

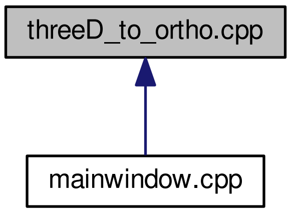
2.5 threeD_to_ortho.cpp File Reference

```
#include <bits/stdc++.h>
#include <armadillo>
#include "input.cpp"
#include <math.h>
```

Include dependency graph for threeD_to_ortho.cpp:



This graph shows which files directly or indirectly include this file:



Macros

- #define PI 3.1415926536

Functions

- dir_ratios get_dir_ratios ()
- mat graph_to_mat (vector< node > nodes, int cols=4)
- vector< node > mat_to_graph (mat A, vector< node > vec)
- mat translate_graph (mat A, coordinate t_factor)
- rot_matrix rot_about_coord_axis (direction theta)
- mat find_rot (mat A, dir_ratios d)
- mat find_projection (mat A)
- vector< node > find_ortho (vector< node > graph)
- void rotate_graph (vector< node > &v, direction theta, int x, int y, int z)

2.5.1 Macro Definition Documentation

2.5.1.1 PI

```
#define PI 3.1415926536
```

2.5.2 Function Documentation

2.5.2.1 find_ortho()

```
vector<node> find_ortho (
            vector< node > graph )
```

Takes the graph whose projection is required to be taken as an argument. Then asks the user about the direction ratios of the desired line of sight. Then converts the graph to a matrix. Finds the transitions to be done. Applies the transitions to the coordinate matrx. Converts the matrix back to a graph and returns it.

2.5.2.2 find_projection()

```
mat find_projection (
            mat A )
```

Returns the projection of the graph after applying all the required transitions using z axis as the line-of-sight.

2.5.2.3 find_rot()

```
mat find_rot (
            mat A,
            dir_ratios d )
```

Takes two inputs

- The matrix A which has to translated
- The direction ratios of the direction which has to become the line of sight for taking the projections. This function calls the function rot_about_coord_axis to get the rotation matrices and then multiplies the matrices with the coordinate matrix. It returns the coordinate matrix after multiplying it with the rotation matrices.

2.5.2.4 get_dir_ratios()

```
dir_ratios get_dir_ratios ( )
```

Asks the user for the direction along which it is desired to take the projection.

2.5.2.5 graph_to_mat()

```
mat graph_to_mat (
            vector< node > nodes,
            int cols = 4 )
```

Takes a graph as an input. Converts the graph into a 4X4 coordinate matrix as specified in the mathematical model.

2.5.2.6 mat_to_graph()

```
vector<node> mat_to_graph (
             mat A,
             vector< node > vec )
```

Takes a coordinate matrix as an input. Converts 4X4 coordinate matrix to the graph which would lead to construiction of this matrix.

2.5.2.7 rot_about_coord_axis()

```
rot_matrix rot_about_coord_axis (
             direction theta )
```

Takes the direction ratios as the input It returns the rotation matrices which are to be multiplied to the coordinate matrix so as to make the Z axis coincide with the given direction theta. It returns a struct of rot_matrix type which contains the three rotation matrices namely Rx, Ry and Rz.

2.5.2.8 rotate_graph()

```
void rotate_graph (
             vector< node > & v,
             direction theta,
             int x,
             int y,
             int z )
```

Rotate the given graph about axis (X, Y or Z axis) depending on the input x, y and z Finsd the rotations matrix for the given direction theta and applies only those whose corresponding input is 1. For Rx, x is to be 1. For Ry, y is to be 1. For Rz, z is to be 1.

2.5.2.9 translate_graph()

```
mat translate_graph (
             mat A,
             coordinate t_factor )
```

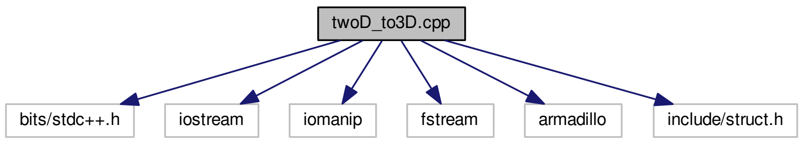
Takes two inputs

- The matrix A (for the corresponding graph) which has to translated
- The coordinate t_factor of the point that would become the origin in the translated system. This function makes multiple calls to the function translate_coordinate for all the vertices of the input graph.

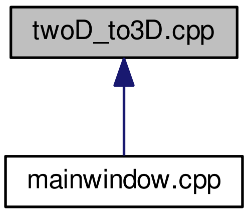
2.6 twoD_to3D.cpp File Reference

```
#include <bits/stdc++.h>
#include <iostream>
#include <iomanip>
#include <fstream>
#include <armadillo>
#include "include/struct.h"
```

Include dependency graph for twoD_to3D.cpp:



This graph shows which files directly or indirectly include this file:



Functions

- vector< node > get_2D_graph (string filename)
- vector< pair_ > get_pair_2D (string filename)
- void check_graph (vector< node > v)

2.6.1 Function Documentation

2.6.1.1 check_graph()

```
void check_graph (
          vector< node > v )
```

2.6.1.2 get_2D_graph()

```
vector<node> get_2D_graph (
             string filename )
```

Reads the 2D input from a file given as an argument. Converts the input into a graph and returns the graph.

2.6.1.3 get_pair_2D()

```
vector<pair_> get_pair_2D (
             string filename )
```

Reads the 2D input from a file given as an argument. Returns all pairs of vertices that have an edge between them according to the input in the file.

Index