

labo_10_comte_emmanuelle_gallay_david

Generated by Doxygen 1.8.13

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	geometry.h File Reference	3
2.1.1	Typedef Documentation	4
2.1.1.1	Coordinate	4
2.1.1.2	Line	4
2.1.1.3	Map	4
2.1.1.4	Point	5
2.1.1.5	PointList	5
2.1.1.6	Space	5
2.1.2	Enumeration Type Documentation	5
2.1.2.1	MapCode	5
2.1.2.2	PointState	5
2.1.3	Function Documentation	5
2.1.3.1	addPoint()	6
2.1.3.2	displayLine()	6
2.1.3.3	displayMap()	6
2.1.3.4	getPointList()	7
2.1.3.5	getProjection()	7
2.1.3.6	getSpace()	7
2.1.3.7	getX()	8
2.1.3.8	getY()	8
2.1.3.9	getZ()	8
2.1.3.10	project()	9
2.1.3.11	setX()	9
2.1.3.12	setY()	9
2.1.3.13	setZ()	10
	Index	11

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

geometry.h	3
--------------------------------------	---

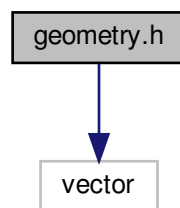
Chapter 2

File Documentation

2.1 geometry.h File Reference

```
#include <vector>
```

Include dependency graph for geometry.h:



Typedefs

- typedef std::size_t [Coordinate](#)
- typedef std::vector< [Coordinate](#) > [Point](#)
- typedef std::vector< [Point](#) > [PointList](#)
- typedef std::vector< [PointState](#) > [Line](#)
- typedef std::vector< [Line](#) > [Map](#)
- typedef std::vector< [Map](#) > [Space](#)

Enumerations

- enum [PointState](#) { [EMPTY](#), [EXIST](#) }
- enum [MapCode](#) { [MAPCODE_XY](#), [MAPCODE_XZ](#), [MAPCODE_YZ](#) }

Functions

- [PointList](#) [getPointList](#) ()
Get a point list from the user.
- void [displayLine](#) (const [Line](#) &line, char exist='O', char empty='.')
- *Display a line.*
- void [displayMap](#) (const [Map](#) &map, [MapCode](#) code, char exist='O', char empty='.')
- *Display a map.*
- [Space](#) [getSpace](#) (const [PointList](#) &list)
Create a space conatin the three projections.
- void [project](#) (const [PointList](#) &list, [Map](#) &map, [MapCode](#) code)
Project a point in a map (XY, XZ or YZ)
- void [addPoint](#) ([Space](#) &space, [Point](#) point)
Add a point to all of projections in space.
- [Map](#) [getProjection](#) (const [Space](#) &space, [MapCode](#) code)
- [Coordinate](#) [getX](#) (const [Point](#) &point)
Get the Coordonate X from a point.
- [Coordinate](#) [getY](#) (const [Point](#) &point)
Get the Coordonate Y from a point.
- [Coordinate](#) [getZ](#) (const [Point](#) &point)
Get the Coordonate Z from a point.
- void [setX](#) ([Point](#) &point, [Coordinate](#) value)
Set the Coordonate X from a point.
- void [setY](#) ([Point](#) &point, [Coordinate](#) value)
Get the Coordonate Y from a point.
- void [setZ](#) ([Point](#) &point, [Coordinate](#) value)
Get the Coordonate Z from a point.

2.1.1 Typedef Documentation

2.1.1.1 Coordonate

```
typedef std::size_t Coordonate
```

2.1.1.2 Line

```
typedef std::vector<PointState> Line
```

2.1.1.3 Map

```
typedef std::vector<Line> Map
```


2.1.1.4 Point

```
typedef std::vector<Coordinate> Point
```

2.1.1.5 PointList

```
typedef std::vector<Point> PointList
```

2.1.1.6 Space

```
typedef std::vector<Map> Space
```

2.1.2 Enumeration Type Documentation

2.1.2.1 MapCode

```
enum MapCode
```

Enumerator

MAPCODE_XY	
MAPCODE_XZ	
MAPCODE_YZ	

2.1.2.2 PointState

```
enum PointState
```

Enumerator

EMPTY	
EXIST	

2.1.3 Function Documentation

2.1.3.1 addPoint()

```
void addPoint (
    Space & space,
    Point point )
```

Add a point to all of projections in space.

Parameters

in	<i>PointList</i>	list
in		

2.1.3.2 displayLine()

```
void displayLine (
    const Line & line,
    char exist = 'O',
    char empty = '.' )
```

Display a line.

Parameters

in	<i>const</i>	Line& line
in	<i>char</i>	exist Display for a point Default value 'O'
in	<i>char</i>	empty Display for a place empty Default value '.'

2.1.3.3 displayMap()

```
void displayMap (
    const Map & map,
    MapCode code,
    char exist = 'O',
    char empty = '.' )
```

Display a map.

Parameters

in	<i>const</i>	Map& map
in	<i>char</i>	exist Display for a point Default value 'O'
in	<i>char</i>	empty Display for a place empty Default value '.'

2.1.3.4 getPointList()

```
PointList getPointList ( )
```

Get a point list from the user.

Returns

PointList The point list the user enter

2.1.3.5 getProjection()

```
Map getProjection (
    const Space & space,
    MapCode code )
```

Parameters

in	<i>const</i>	Space& space
in	<i>MapCode</i>	code The code for the direction of the projection

Returns

Map The map of the projection asked

2.1.3.6 getSpace()

```
Space getSpace (
    const PointList & list )
```

Create a space conatin the three projections.

Parameters

in	<i>PointList</i>	list The list of the points in the space
----	------------------	--

Returns

Space Contain the three projections

2.1.3.7 getX()

```
Coordinate getX (
    const Point & point )
```

Get the Coordonate X from a point.

Parameters

in	<i>Point</i>	point
----	--------------	-------

Returns

Coordonate The Coordonate X of the point

2.1.3.8 getY()

```
Coordinate getY (
    const Point & point )
```

Get the Coordonate Y from a point.

Parameters

in	<i>Point</i>	point
----	--------------	-------

Returns

Coordonate The Coordonate Y of the point

2.1.3.9 getZ()

```
Coordinate getZ (
    const Point & point )
```

Get the Coordonate Z from a point.

Parameters

in	<i>Point</i>	point
----	--------------	-------

Returns

Coordonate The Coordonate Z of the point

2.1.3.10 project()

```
void project (
    const PointList & list,
    Map & map,
    MapCode code )
```

Project a point in a map (XY, XZ or YZ)

Parameters

in	<i>PointList</i>	list
in		

2.1.3.11 setX()

```
void setX (
    Point & point,
    Coordinate value )
```

Set the Coordonate X from a point.

Parameters

in	<i>Point</i>	point
----	--------------	-------

2.1.3.12 setY()

```
void setY (
    Point & point,
    Coordinate value )
```

Get the Coordonate Y from a point.

Parameters

in	<i>Point</i>	point
----	--------------	-------

2.1.3.13 setZ()

```
void setZ (
    Point & point,
    Coordonate value )
```

Get the Coordonate Z from a point.

Parameters

in	<i>Point</i>	point
----	--------------	-------

Index

addPoint
 [geometry.h, 5](#)

Coordonate
 [geometry.h, 4](#)

displayLine
 [geometry.h, 6](#)

displayMap
 [geometry.h, 6](#)

[geometry.h, 3](#)
 addPoint, [5](#)
 Coordonate, [4](#)
 displayLine, [6](#)
 displayMap, [6](#)
 getPointList, [6](#)
 getProjection, [7](#)
 getSpace, [7](#)
 getX, [7](#)
 getY, [8](#)
 getZ, [8](#)
 Line, [4](#)
 Map, [4](#)
 MapCode, [5](#)
 Point, [4](#)
 PointList, [5](#)
 PointState, [5](#)
 project, [9](#)
 setX, [9](#)
 setY, [9](#)
 setZ, [9](#)
 Space, [5](#)

getPointList
 [geometry.h, 6](#)

getProjection
 [geometry.h, 7](#)

getSpace
 [geometry.h, 7](#)

getX
 [geometry.h, 7](#)

getY
 [geometry.h, 8](#)

getZ
 [geometry.h, 8](#)

Line
 [geometry.h, 4](#)

Map
 [geometry.h, 4](#)

MapCode
 [geometry.h, 5](#)

Point
 [geometry.h, 4](#)

PointList
 [geometry.h, 5](#)

PointState
 [geometry.h, 5](#)

project
 [geometry.h, 9](#)

setX
 [geometry.h, 9](#)

setY
 [geometry.h, 9](#)

setZ
 [geometry.h, 9](#)

Space
 [geometry.h, 5](#)