

labo\_10\_comte\_emmanuelle\_gallay\_david

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>File Index</b>	<b>1</b>
1.1	File List . . . . .	1
<b>2</b>	<b>File Documentation</b>	<b>3</b>
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 . . . . .	4
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() . . . . .	5
2.1.3.2	displayLine() . . . . .	7
2.1.3.3	displayMap() . . . . .	7
2.1.3.4	getPointList() . . . . .	7
2.1.3.5	getProjection() . . . . .	7
2.1.3.6	getSpace() . . . . .	8
2.1.3.7	getX() . . . . .	8
2.1.3.8	getY() . . . . .	9
2.1.3.9	getZ() . . . . .	9
2.1.3.10	project() . . . . .	9
	<b>Index</b>	<b>11</b>



# Chapter 1

## File Index

### 1.1 File List

Here is a list of all files with brief descriptions:

<a href="#">geometry.h</a> . . . . .	3
--------------------------------------	---



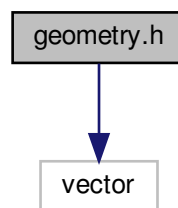
## Chapter 2

# 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)  
*Display a line.*
- void [displayMap](#) (const [Map](#) &map)  
*Display a map.*
- [Space](#) [getSpace](#) ([PointList](#) list)  
*Create a space conatin the three projections.*
- void [project](#) ([PointList](#) list, [Map](#) &map, [MapCode](#) code)  
*Project a point in a map (XY, XZ or YZ)*
- void [addPoint](#) ([Point](#) point, [Space](#) &space)  
*Add a point to all of projections in space.*
- [Map](#) [getProjection](#) (const [Space](#) &space, [MapCode](#) code)
- [Coordinate](#) [getX](#) ([Point](#) point)  
*Get the coordinate X from a point.*
- [Coordinate](#) [getY](#) ([Point](#) point)  
*Get the coordinate Y from a point.*
- [Coordinate](#) [getZ](#) ([Point](#) point)  
*Get the coordinate Z from a point.*

## 2.1.1 Typedef Documentation

### 2.1.1.1 Coordinate

```
typedef std::size_t Coordinate
```

### 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 (
    Point point,
    Space & space )
```

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 )
```

Display a line.

**Parameters**

in	<i>const</i>	Line& line
----	--------------	------------

**2.1.3.3 displayMap()**

```
void displayMap (
    const Map & map )
```

Display a map.

**Parameters**

in	<i>const</i>	Map& map
----	--------------	----------

**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 (
    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 (
    Point point )
```

Get the coordinate X from a point.

**Parameters**

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

**Returns**

Coordinate The coordinate X of the point

### 2.1.3.8 getY()

```
Coordinate getY (
    Point point )
```

Get the coordinate Y from a point.

#### Parameters

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

#### Returns

Coordinate The coordinate Y of the point

### 2.1.3.9 getZ()

```
Coordinate getZ (
    Point point )
```

Get the coordinate Z from a point.

#### Parameters

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

#### Returns

Coordinate The coordinate Z of the point

### 2.1.3.10 project()

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

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

#### Parameters

in	<i>PointList</i>	list
in		



# Index

addPoint  
    [geometry.h, 5](#)

Coordinate  
    [geometry.h, 4](#)

displayLine  
    [geometry.h, 7](#)

displayMap  
    [geometry.h, 7](#)

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

getPointList  
    [geometry.h, 7](#)

getProjection  
    [geometry.h, 7](#)

getSpace  
    [geometry.h, 8](#)

getX  
    [geometry.h, 8](#)

getY  
    [geometry.h, 8](#)

getZ  
    [geometry.h, 9](#)

Line  
    [geometry.h, 4](#)

Map  
    [geometry.h, 4](#)

MapCode  
    [geometry.h, 5](#)

Point  
    [geometry.h, 4](#)

PointList  
    [geometry.h, 4](#)

PointState  
    [geometry.h, 5](#)

project  
    [geometry.h, 9](#)

Space  
    [geometry.h, 5](#)