

vector3d
56387

Generated by Doxygen 1.5.5

Fri Apr 25 11:29:02 2008

Contents

Chapter 1

Class Index

1.1 Class List

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

[vector3d](#) (Basic 3D-vector class with fundamental vector operations) ??

Chapter 2

Class Documentation

2.1 vector3d Class Reference

Basic 3D-vector class with fundamental vector operations.

```
#include <vector3d.h>
```

Public Member Functions

- `vector3d ()`
Default constructor constructs a zero- vector.
- `vector3d (double X, double Y, double Z)`
Constructor with coordinates.
- `double operator[] (int i) const`
Return coordinates via index (0,1,2).
- `double & operator[] (int i)`
Acces vector coordinates via index mode (0,1,2).
- `operator const double * () const`
Returning the memory address of the vector.
- `vector3d & operator= (const vector3d &v)`
= operator.
- `vector3d & operator/= (double k)`
Divides the vector by k.
- `double norm () const`
Calculates the norm of the vector.
- `double normalize ()`
Normalizes the vector.

Static Public Member Functions

- static double `vecAngle` (const `vector3d` &a, const `vector3d` &b)
angle (degrees) between vector a and vector b
- static double `vecRadAngle` (const `vector3d` &a, const `vector3d` &b)
angle (radians) between vector a and vector b

Public Attributes

- union {
 struct {
 double **x**
 double **y**
 double **z**
 }
 double **c** [3]
};

Friends

- `vector3d operator-` (const `vector3d` &a, const `vector3d` &b)
Subtracts vector b from vector a.
- double `operator*` (const `vector3d` &a, const `vector3d` &b)
Dot product of vector a and vector b.
- `vector3d operator*` (const `vector3d` &a, float k)
Returns the product of the vector a with a scalar k.
- `vector3d operator*` (float k, const `vector3d` &a)
Returns the product of the vector a with a scalar k.
- `vector3d operator+` (const `vector3d` &a, const `vector3d` &b)
Sum of vector a and vector b.
- `vector3d operator/` (const `vector3d` &a, double k)
Division of vector a by k.
- bool `operator==` (const `vector3d` &a, const `vector3d` &b)
Checks whether two vectors are equal.
- `vector3d cross` (const `vector3d` &a, const `vector3d` &b)
crossproduct of vector a and vector b

2.1.1 Detailed Description

Basic 3D-vector class with fundamental vector operations.

2.1.2 Constructor & Destructor Documentation

2.1.2.1 `vector3d::vector3d ()` [inline]

Default constructor constructs a zero- vector.

2.1.3 Member Function Documentation

2.1.3.1 `double& vector3d::operator[] (int i)` [inline]

Acces vector coordinates via index mode (0,1,2).

(lvalue)

2.1.3.2 `vector3d::operator const double * () const` [inline]

Returning the memory address of the vector.

Useful for passing the vector to openGL functions

2.1.3.3 `vector3d& vector3d::operator= (const vector3d & v)` [inline]

= operator.

2.1.3.4 `vector3d& vector3d::operator/= (double k)` [inline]

Divides the vector by k.

*

```
vector3d& operator/=(float k) { x /=k; y /=k; z /=k; return *this; }
```

Divides the vector by k.

2.1.3.5 `double vector3d::norm () const` [inline]

Calculates the norm of the vector.

$$\sqrt{x^2 + y^2 + z^2}$$

2.1.3.6 `double vector3d::normalize ()` [inline]

Normalizes the vector.

If the vector is a null vector (`norm()` < `vector3d_ZERO`) the vector is not modified and function returns -1.

Otherwise the former norm of the vector is returned.

2.1.3.7 static double vector3d::vecAngle (const vector3d & a, const vector3d & b) [inline, static]

angle (degrees) between vector a and vector b

$$\alpha = \arccos\left(\frac{\vec{a} * \vec{b}}{|\vec{a}| * |\vec{b}|}\right)$$

2.1.3.8 static double vector3d::vecRadAngle (const vector3d & a, const vector3d & b) [inline, static]

angle (radians) between vector a and vector b

$$\alpha = \arccos\left(\frac{\vec{a} * \vec{b}}{|\vec{a}| * |\vec{b}|}\right)$$

2.1.4 Friends And Related Function Documentation

2.1.4.1 double operator* (const vector3d & a, const vector3d & b) [friend]

Dot product of vector a and vector b.

2.1.4.2 vector3d operator* (const vector3d & a, float k) [friend]

Returns the product of the vector a with a scalar k.

2.1.4.3 vector3d operator* (float k, const vector3d & a) [friend]

Returns the product of the vector a with a scalar k.

2.1.4.4 vector3d operator/ (const vector3d & a, double k) [friend]

Division of vector a by k.

friend [vector3d](#) operator/(const vector3d &a, float k) { return [vector3d](#)(a.x/k, a.y/k, a.z/k); }

Division of vector a by k.

2.1.4.5 bool operator== (const vector3d & a, const vector3d & b) [friend]

Checks whether two vectors are equal.

The two vectors are equal, If the norm of the difference of the two vectors is smaller than `vector3d_ZERO`

2.1.4.6 vector3d cross (const vector3d & a, const vector3d & b) [friend]

crossproduct of vector a and vector b

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

- `vector3d.h`