Geometrics

Generated by Doxygen 1.7.6.1

Wed Mar 5 2014 23:01:28

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

Class Documentation

2.1 Quaternion Class Reference

Public Member Functions

- **Quaternion** (float inW, float inX, float inY, float inZ)
- **Quaternion** (float alpha, float beta, float gamma)
- template<typename T >
  Quaternion (float angle, Vec3< T > const &axis)
- template<typename T , typename U >
  Quaternion (Vec3< T > const &v1, Vec3< U > const &v2)
- Quaternion **operator*** (Quaternion const &rOp) const
- Quaternion **operator+** (Quaternion const &rOP) const
- void **normalize** ()
- bool **isNormalized** () const
- float **angle** (Quaternion const &toQuat) const
- Quaternion **slerp** (Quaternion const &destQt, float t, float eps=0.01) const
- Quaternion **lerp** (Quaternion const &destQt, float t) const
- void **toByteArray** (byte *bArray) const
- float **rotAngleInDeg** ()

Public Attributes

- float **w**
- float **x**
- float **y**
- float **z**

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

- Geometrics/Quaternion.h

2.2 Vec3< T > Struct Template Reference

Public Member Functions

- **Vec3** (T inX, T inY, T inZ)
- float **norm2** () const
- template<typename U >
  U **dot** (Vec3< U > const &v) const
- template<typename U >
  Vec3< U > **cross** (Vec3< U > const &v) const

Public Attributes

- T **x**
- T **y**
- T **z**

template<typename T> struct Vec3< T >

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

- Geometrics/Vec3.h

2.3 Geometrics::Vector< T > Class Template Reference

Public Member Functions

- Vector (T coordinates[], const int dimension)
- Vector (const int dim, const T value)
- virtual ~Vector ()
- bool operator== (const Vector &v)
- bool operator!= (const Vector &v)
- const Vector operator+ (const Vector &v)
- const Vector operator- (const Vector &v)
- T & operator[] (const int &i)
- const T & operator[] (const int &i) const

template<class T = int> class Geometrics::Vector< T >

2.3.1 Constructor & Destructor Documentation

2.3.1.1 template<class T = int> Geometrics::Vector< T >::Vector (T *coordinates[]*, const int *dimension*) [inline]

The first constructor.

Parameters

coordinates	The coordinates of the Vector.
dimension	The dimension of the Vector.

2.3.1.2 template<class T = int> Geometrics::Vector< T >::Vector (const int *dim*, const T *value*) [inline]

The second constructor

Parameters

dim	The dimension of the Vector.
value	All coordinates are set to that value.

2.3.1.3 template<class T = int> virtual Geometrics::Vector< T >::~Vector ( ) [inline, virtual]

The destructor, which deletes the array, storing the coordinates.

2.3.2 Member Function Documentation

2.3.2.1 template<class T = int> bool Geometrics::Vector< T >::operator!= (const Vector< T > & *v*) [inline]

Overloading the != operator.

Parameters

v	The other Vector.

Returns

True, if not all the coordinates of both Vector are equal.

2.3.2.2 template<class T = int> const Vector Geometrics::Vector< T >::operator+ (const Vector< T > & *v*) [inline]

Overloading the + operator. Add two vector v1 and v2. Throw an assertion, if the dimension of the vectors are not the same.

Parameters

v	The other Vector.

Returns

Vector v3, where all coordinate i holds: v3[i] = v1[i] + v2[i].

2.3.2.3 template<class T = int> const Vector Geometrics::Vector< T >::operator- (const Vector< T > & *v*) [inline]

Overloading the - operator. Add two vector v1 and v2. Throw an assertion, if the dimension of the vectors are not the same.

Parameters

v	The other Vector.

Returns

Vector v3, where all coordinate i holds: v3[i] = v1[i] - v2[i].

2.3.2.4 template<class T = int> bool Geometrics::Vector< T >::operator== (const Vector< T > & *v*) [inline]

Overloading the == operator.

Parameters

v	The other Vector.

Returns

True, if all the coordinates of both Vectors are the same.

2.3.2.5 template<class T = int> T& Geometrics::Vector< T >::operator[] (const int & *i*) [inline]

Overloading the [] operator. Non-Const variante.

Parameters

i	is the coordinate index

Returns

The value of the coordinate with the index i.

2.3.2.6 template<class T = int> const T& Geometrics::Vector< T >::operator[ ] (const int & *i*) const [inline]

Overloading the [] operator. Const variante.

Parameters

i	is the coordinate index

Returns

The value of the coordinate with the index i.

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

- Geometrics/Vector.h