Polylib

3.1.0

Generated by Doxygen 1.8.5

Wed Nov 13 2013 16:24:33

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 PolylibNS Namespace Reference

Classes

- class BBox
- class PolylibStat2
- class Vec2
- class Vec3
- class TriMeshIO
- class TryMesh
- class Polylib
- class Polygons
- class PolygonGroup
- class PolygonGroupFactory
- struct ParallelInfo
- class MPIPolylib
- class DVertex
- class DVertexManager
- class DVertexTriangle
- class Triangle
- class PrivateTriangle
- class VTree
- class VertexList
- class VertKDT
- struct PrivTriaLess
- struct PrivTriaEqual
- class TriMesh
- class Vertex
- class VertKDTElem
- class VertKDTNode
- class VElement
- class VNode
- struct CalcAreaInfo
- class PolylibMoveParams

Typedefs

- typedef Vec2< unsigned char > **Vec2uc**
- typedef Vec2< int > **Vec2i**
- typedef Vec2< float > **Vec2f**
- typedef Vec3< unsigned char > **Vec3uc**
- typedef Vec3< int > **Vec3i**
- typedef Vec3< float > **Vec3f**
- typedef Vec3< double > **Vec3d**
- typedef unsigned int **uint**
- typedef unsigned short **ushort**

Enumerations

- enum ID_FORMAT { ID_BIN, ID_ASCII }

Functions

- template<typename T >
  Vec2< T > **operator**∗ (T s, const Vec2< T > &v)
- template<typename T >
  T **distanceSquared** (const Vec2< T > &a, const Vec2< T > &b)
- template<typename T >
  T **distance** (const Vec2< T > &a, const Vec2< T > &b)
- template<typename T >
  bool **lessVec2** (const Vec2< T > &a, const Vec2< T > &b)
- template<typename T >
  std::istream & **operator**>> (std::istream &is, Vec2< T > &v)
- template<typename T >
  std::ostream & **operator**<< (std::ostream &os, const Vec2< T > &v)
- std::istream & **operator**>> (std::istream &is, Vec2uc &v)
- std::ostream & **operator**<< (std::ostream &os, const Vec2uc &v)
- bool **lessVec2f** (const Vec2f &a, const Vec2f &b)
- template<typename T >
  Vec3< T > **operator**∗ (T s, const Vec3< T > &v)
- template<typename T >
  Vec3< T > **multi** (const Vec3< T > &a, const Vec3< T > &b)
- template<typename T >
  T **dot** (const Vec3< T > &a, const Vec3< T > &b)
- template<typename T >
  Vec3< T > **cross** (const Vec3< T > &a, const Vec3< T > &b)
- template<typename T >
  T **distanceSquared** (const Vec3< T > &a, const Vec3< T > &b)
- template<typename T >
  T **distance** (const Vec3< T > &a, const Vec3< T > &b)
- template<typename T >
  std::istream & **operator**>> (std::istream &is, Vec3< T > &v)
- template<typename T >
  std::ostream & **operator**<< (std::ostream &os, const Vec3< T > &v)
- std::istream & **operator**>> (std::istream &is, Vec3uc &v)
- std::ostream & **operator**<< (std::ostream &os, const Vec3uc &v)
- bool **lessVec3f** (const Vec3f &a, const Vec3f &b)
- template<typename T1 , typename T2 >
  void **vec3_copy** (T1 to[3], const T2 from[3])

- template<typename T1 , typename T2 >
  void **vec3_set** (T1 v[3], T2 x, T2 y, T2 z)
- template<typename T >
  void **vec3_min** (T c[3], const T a[3], const T b[3])
- template<typename T >
  void **vec3_max** (T c[3], const T a[3], const T b[3])
- template<typename T1 , typename T2 , typename T3 >
  void **vec3_plus** (T1 c[3], const T2 a[3], const T3 b[3])
- template<typename T1 , typename T2 , typename T3 >
  void **vec3_minus** (T1 c[3], const T2 a[3], const T3 b[3])
- template<typename T1 , typename T2 , typename T3 >
  void **vec3_multi** (T1 c[3], const T2 a[3], const T3 b[3])
- template<typename T1 , typename T2 , typename T3 >
  void **vec3_multi** (T1 c[3], const T2 a[3], const T3 b)
- template<typename T1 , typename T2 , typename T3 >
  void **vec3_div** (T1 c[3], const T2 a[3], const T3 b[3])
- template<typename T1 , typename T2 , typename T3 >
  void **vec3_div** (T1 c[3], const T2 a[3], const T3 b)
- void **vec3f_copy** (float to[3], const float from[3])
- void **vec3f_set** (float v[3], float x, float y, float z)
- void **vec3f_min** (float c[3], const float a[3], const float b[3])
- void **vec3f_max** (float c[3], const float a[3], const float b[3])
- void **vec3f_plus** (float c[3], const float a[3], const float b[3])
- void **vec3f_minus** (float c[3], const float a[3], const float b[3])
- void **vec3f_multi** (float c[3], const float a[3], const float b[3])
- void **vec3f_multi** (float c[3], const float a[3], const float b)
- void **vec3f_div** (float c[3], const float a[3], const float b[3])
- void **vec3f_div** (float c[3], const float a[3], const float b)
- float **vec3f_dot** (const float a[3], const float b[3])
- void **vec3f_cross** (float c[3], const float a[3], const float b[3])
- float **vec3f_sqdist** (const float a[3], const float b[3])
- float **vec3f_dist** (const float a[3], const float b[3])
- bool is_obj_a (std::string path)
- template<typename T >
  POLYLIB_STAT obj_a_load (VertexList< T > *vertex_list, std::vector< PrivateTriangle< T > * > *tri_list, std::string fname, int *total, T scale=1.0)
- template<typename T >
  POLYLIB_STAT obj_b_load (VertexList< T > *vertex_list, std::vector< PrivateTriangle< T > * > *tri_list, std::string fname, int *total, T scale=1.0)
- template<typename T >
  POLYLIB_STAT obj_a_save (VertexList< T > *vertex_list, std::vector< PrivateTriangle< T > * > *tri_list, std::string fname)
- template<typename T >
  POLYLIB_STAT obj_b_save (VertexList< T > *vertex_list, std::vector< PrivateTriangle< T > * > *tri_list, std::string fname)
- template<typename T >
  POLYLIB_STAT obj_bb_save (VertexList< T > *vertex_list, std::vector< PrivateTriangle< T > * > *tri_list, std::string fname)
- template<typename T >
  POLYLIB_STAT stl_a_load (VertexList< T > *vertex_list, std::vector< PrivateTriangle< T > * > *tri_list, std::string fname, int *total, T scale=1.0)
- template<typename T >
  POLYLIB_STAT stl_a_save (std::vector< PrivateTriangle< T > * > *tri_list, std::string fname)
- template<typename T >
  POLYLIB_STAT stl_b_load (VertexList< T > *vertex_list, std::vector< PrivateTriangle< T > * > *tri_list, std::string fname, int *total, T scale=1.0)

- template<typename T >
  POLYLIB_STAT stl_b_save (std::vector< PrivateTriangle< T > * > *tri_list, std::string fname)
- bool is_stl_a (std::string path)
- char * stl_get_fname (std::string path)
- char * stl_get_ext (std::string path)
- template<typename T >
  POLYLIB_STAT load_id (std::vector< PrivateTriangle< T > * > *tri_list, std::string fname, ID_FORMAT id_format)
- template<typename T >
  POLYLIB_STAT save_id (std::vector< PrivateTriangle< T > * > *tri_list, std::string fname, ID_FORMAT id_format)
- template<typename T >
  POLYLIB_STAT vtk_a_save (VertexList< T > *vertex_list, std::vector< PrivateTriangle< T > * > *tri_list, std::string fname)
- template<typename T >
  POLYLIB_STAT vtk_b_save (VertexList< T > *vertex_list, std::vector< PrivateTriangle< T > * > *tri_list, std::string fname)
- bool **getrusage_sec** (double *usr_time, double *sys_time, double *total)

Variables

- std::string gs_rankno

5.1.1 Detailed Description

クラス:DVertexTriangle クラス DVertex を頂点に持つTriagnle クラス

5.1.2 Enumeration Type Documentation

5.1.2.1 enum PolylibNS::ID_FORMAT

三角形ID ファイルフォーマット

Enumerator

ID_BIN バイナリ形式で入出力を行う。

ID_ASCII アスキー形式で入出力を行う。

5.1.3 Function Documentation

5.1.3.1 bool PolylibNS::is_obj_a (std::string *path*)

ファイルの一部を読み込み、ascii / binary を判定する。

Parameters

in	*path*	ファイルパス

5.1.3.2 bool PolylibNS::is_stl_a (std::string *path*)

STL ファイルを読み込みバイナリかアスキーかを判定する。

Parameters

in	*STL* ファイルのフルパス名。	

Returns

true:アスキー形式 / false:バイナリ形式。

5.1.3.3 template<typename T > POLYLIB_STAT PolylibNS::load_id (std::vector< PrivateTriangle< T > * > * *tri_list*, std::string *fname*, ID_FORMAT *id_format*)

三角形ポリゴンID をID ファイルから読み込み、m_id にセットする。

Parameters

in,out	*tri_list*	三角形ポリゴン情報。
in	*fname*	三角形ポリゴンID ファイル名。
in	*id_format*	三角形ポリゴンID ファイルの入力形式。

Returns

POLYLIB_STAT で定義される値が返る

5.1.3.4 template<typename T > POLYLIB_STAT PolylibNS::obj_a_load (VertexList< T > * *vertex_list*, std::vector< PrivateTriangle< T > * > * *tri_list*, std::string *fname*, int * *total*, T *scale* = 1.0)

ASCII モードのOBJ ファイルを読み込み、VertexList, tri_list に三角形ポリゴン情報を設定する。

Parameters

in,out	*vertex_list*	頂点リストの領域。
in,out	*tri_list*	三角形ポリゴンリストの領域。
in	*fname*	STL ファイル名。
in,out	*total*	ポリゴンID の通番。

Returns

POLYLIB_STAT で定義される値が返る。

エラーについて

1. ファイルが開けないとき
2. face のリストがまだ読み込まれていない頂点ID を使った場合

注意事項 face はすべて三角形だとして読み込む。 情報として取り込むのは、v と f のみで、他の情報は破棄される。 v と f の情報から、normal を計算する。

5.1.3.5 template<typename T > POLYLIB_STAT PolylibNS::obj_a_save (VertexList< T > * *vertex_list*, std::vector< PrivateTriangle< T > * > * *tri_list*, std::string *fname*)

VertexList, tri_list にから OBJ アスキー形式に出力する。

Parameters

in	*vertex_list*	頂点リストの領域。
in	*tri_list* **三角形ポリゴンリストの領域。**	
in	*fname*	ファイル名。

エラーについて

注意事項 情報として書き出すのは、v, vn と f のみで、他の情報は破棄される。 すべての面を持たない頂点の頂点法線は、正しく計算されない可能性があるので、注意すること。

5.1.3.6 template<typename T > POLYLIB_STAT PolylibNS::obj_b_load (VertexList< T > * *vertex_list*, std::vector< PrivateTriangle< T > * > * *tri_list*, std::string *fname*, int * *total*, T *scale* = 1.0)

OBJ_BIN 形式のファイルを読み込み、 VertexList, tri_list に三角形ポリゴン情報を設定する。 頂点法線が記録されているかどうかを判別して読み取る。 頂点法線は記録されていても、この時点で情報を捨てる。

Parameters

in,out	*vertex_list*	頂点リストの領域。
in,out	*tri_list*	三角形ポリゴンリストの領域。
in	*fname*	STL ファイル名。
in,out	*total*	ポリゴンID の通番。

Returns

POLYLIB_STAT で定義される値が返る。

エラーについて

1. ファイルが開けないとき
2. face のリストがまだ読み込まれていない頂点ID を使った場合

注意事項 face はすべて三角形だとして読み込む。 情報として取り込むのは、v と f のみで、他の情報は破棄される。 v と f の情報から、normal を計算する。

5.1.3.7 template<typename T > POLYLIB_STAT PolylibNS::obj_b_save (VertexList< T > * *vertex_list*, std::vector< PrivateTriangle< T > * > * *tri_list*, std::string *fname*)

VertexList, tri_list にから OBJ バイナリ形式に出力する。

Parameters

in	*vertex_list*	頂点リストの領域。
in	*tri_list* **三角形ポリゴンリストの領域。**	
in	*fname*	ファイル名。

エラーについて

注意事項 情報として書き出すのは、v と f のみで、他の情報は破棄される。 #undef DEBUG

#undef DEBUG

5.1.3.8 template<typename T > POLYLIB_STAT PolylibNS::obj_bb_save (VertexList< T > * *vertex_list*, std::vector< PrivateTriangle< T > * > * *tri_list*, std::string *fname*)

VertexList, tri_list にから OBJ バイナリ形式に出力する。

Parameters

in	*vertex_list*	頂点リストの領域。
in	*tri_list* **三角形ポリゴンリストの領域。**	
in	*fname*	ファイル名。

エラーについて

注意事項 情報として書き出すのは、v, vn と f のみで、他の情報は破棄される。 すべての面を持たない頂点の頂点法線は、正しく計算されない可能性があるので、注意すること。

5.1.3.9 template<typename T > POLYLIB_STAT PolylibNS::save_id (std::vector< PrivateTriangle< T > * > * *tri_list*, std::string *fname*, ID_FORMAT *id_format*)

三角形ポリゴンID をファイルに出力する。

Parameters

in	*tri_list*	三角形ポリゴン情報。
in	*fname*	三角形ポリゴンID ファイル名。
in	*id_format*	三角形ポリゴンID ファイルの出力形式。

Returns

POLYLIB_STAT で定義される値が返る

5.1.3.10 template<typename T > POLYLIB_STAT PolylibNS::stl_a_load (VertexList< T > * *vertex_list*, std::vector< PrivateTriangle< T > * > * *tri_list*, std::string *fname*, int * *total*, T *scale* = 1.0)

ASCII モードのSTL ファイルを読み込み、VertexList, tri_list に三角形ポリゴン情報を設定する。

Parameters

in,out	*vertex_list*	頂点リストの領域。
in,out	*tri_list*	三角形ポリゴンリストの領域。
in	*fname*	STL ファイル名。
in,out	*total*	ポリゴンID の通番。

Returns

POLYLIB_STAT で定義される値が返る。

5.1.3.11 template<typename T > POLYLIB_STAT PolylibNS::stl_a_save (std::vector< PrivateTriangle< T > * > * *tri_list*, std::string *fname*)

三角形ポリゴン情報をASCII モードでSTL ファイルに書き出す。

Parameters

in	*tri_list*	三角形ポリゴン情報。
in	*fname*	STL ファイル名。

Returns

POLYLIB_STAT で定義される値が返る。

5.1.3.12 template<typename T > POLYLIB_STAT PolylibNS::stl_b_load (VertexList< T > * *vertex_list*, std::vector< PrivateTriangle< T > * > * *tri_list*, std::string *fname*, int * *total*, T *scale* = 1.0)

バイナリモードのSTL ファイルを読み込み、tri_list に三角形ポリゴン情報を設定 する。

Parameters

in,out	*tri_list*	三角形ポリゴンリストの領域。
in	*fname*	ファイル名。
in,out	*total*	ポリゴンID の通番。

Returns

POLYLIB_STAT で定義される値が返る。

5.1.3.13 template<typename T > POLYLIB_STAT PolylibNS::stl_b_save (std::vector< PrivateTriangle< T > * > * *tri_list*, std::string *fname*)

三角形ポリゴン情報をバイナリモードでSTL ファイルに書き出す。

Parameters

in	*tri_list*	三角形ポリゴン情報。
in	*fname*	STL ファイル名。

Returns

POLYLIB_STAT で定義される値が返る。

5.1.3.14 char* PolylibNS::stl_get_ext (std::string *path*)

STL ファイル名から拡張子のみを取得する。

Parameters

in	*STL* ファイルのフルパス名。	

Returns

拡張子。

Attention

戻り値の char * は解放不要。

5.1.3.15 char* PolylibNS::stl_get_fname (std::string *path*)

STL ファイル名から名称 (拡張子を除いた部分) を取得する。

Parameters

in	*STL* ファイルのフルパス名。	

Returns

拡張子を除いた名称。

Attention

戻り値の char * は解放不要。

5.1.3.16 template<typename T > POLYLIB_STAT PolylibNS::vtk_a_save (VertexList< T > * *vertex_list*, std::vector< PrivateTriangle< T > * > * *tri_list*, std::string *fname*)

VertexList, tri_list にから vtk アスキー形式に出力する。

Parameters

in	*vertex_list*	頂点リストの領域。
in	*tri_list* 三角形ポリゴンリストの領域。	
in	*fname*	ファイル名。

T は、頂点データの実数型

5.1.3.17 template<typename T > POLYLIB_STAT PolylibNS::vtk_b_save (VertexList< T > * *vertex_list*, std::vector< PrivateTriangle< T > * > * *tri_list*, std::string *fname*)

VertexList, tri_list にから vtk バイナリ形式に出力する。

Parameters

in	*vertex_list*	頂点リストの領域。
in	*tri_list* 三角形ポリゴンリストの領域。	
in	*fname*	ファイル名。

T は、頂点データの実数型 DT は、頂点データに付随するスカラーデータの型

5.1.4 Variable Documentation

5.1.4.1 std::string PolylibNS::gs_rankno

デバッグ出力用ランク番号グローバル文字列

Chapter 6

Class Documentation

6.1 PolylibNS::BBox< T > Class Template Reference

`#include <BBox.h>`

Public Member Functions

- **BBox** (T _minx, T _miny, T _minz, T _maxx, T _maxy, T _maxz)
- **BBox** (T _min[3], T _max[3])
- **BBox** (const Vec3< T > &_min, const Vec3< T > &_max)
- void **init** ()
- void **setMinMax** (const Vec3< T > &_min, const Vec3< T > &_max)
- void **add** (const Vec3< T > &v)
- Vec3< T > **getPoint** (int idx) const
- Vec3< T > **center** () const
- Vec3< T > **size** () const
- T **xsize** () const
- T **ysize** () const
- T **zsize** () const
- T **length** (const AxisEnum &axis) const
- T **diameter** () const
- AxisEnum **getMaxAxis** (T &length) const
- bool contain (const Vec3< T > &pos) const
- bool crossed (const BBox< T > &bbox) const
- BBox< T > getCrossedRegion (BBox< T > &other_bbox) const
- Vec2< T > vec3to2 (int axis_id, Vec3< T > &v3) const
- void getFace (int axis_id, Vec3< T > face[2][2]) const
- void getSide (int axis_id, Vec3< T > side[4][2]) const

Public Attributes

- Vec3< T > **min**
- Vec3< T > **max**

6.1.1 Detailed Description

template<typename T>class PolylibNS::BBox< T >

クラス:BBox Bounding Box を管理するクラス

6.1.2 Member Function Documentation

6.1.2.1 template<typename T> bool PolylibNS::BBox< T >::contain (const Vec3< T > & *pos*) const [inline]

引数で与えられた点が、このBBox に含まれるかを判定する。

Parameters

in	*pos*	試行する点

Returns

含まれる場合は true。他は false。

6.1.2.2 template<typename T> bool PolylibNS::BBox< T >::crossed (const BBox< T > & *bbox*) const [inline]

BBox とBBox の交差判定を行う。 KD-Tree の交差判定と同じ。

Parameters

in	*bbox*	試行するBBox

Returns

交差する場合は true。他は false。

6.1.2.3 template<typename T> BBox<T> PolylibNS::BBox< T >::getCrossedRegion (BBox< T > & *other_bbox*) const [inline]

BBox とBBox の重複領域の抽出を行う。 自身の面と他方の辺との交差判定を行う。

Parameters

in	*other_bbox*	試行するBBox

Returns

交差する場合は true。他は false。

6.1.2.4 template<typename T> void PolylibNS::BBox< T >::getFace (int *axis_id,* Vec3< T > *face[2][2]*) const [inline]

引数 axis_id(0=x,1=y,z=2) に垂直な、このBBox の面の対角点を返す。

Parameters

in	*axis_id*	軸番号。0=x 軸、1=y 軸、2=z 軸。
in	*face*	BBox の面の中で、軸に垂直な面の対角点。

6.1.2.5 template<typename T> void PolylibNS::BBox< T >::getSide (int *axis_id,* Vec3< T > *side[4][2]*) const [inline]

引数 axis_id(0=x,1=y,z=2) に平行な、このBBox の辺の端点を返す。

Parameters

in	*axis_id*	軸番号。0=x 軸、1=y 軸、2=z 軸。
in	*side*	BBox の辺の中で、軸に平行な辺の端点。

6.1.2.6 template<typename T> Vec2<T> PolylibNS::BBox< T >::vec3to2 (int *axis_id*, Vec3< T > & *v3*) const [inline]

引数 axis_id(0=x,1=y,z=2) に垂直な成分を詰めて返す。

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/common/BBox.h

6.2 PolylibNS::CalcAreaInfo< T > Struct Template Reference

`#include <Polylib.h>`

Public Attributes

- Vec3< T > m_bpos
  基点座標
- Vec3< T > m_bbsize
  計算領域のボクセル数
- Vec3< T > m_gcsize
  ガイドセルのボクセル数
- Vec3< T > m_dx
  ボクセル 1 辺の長さ
- Vec3< T > m_gcell_min
  ガイドセルを含めた担当領域の最小位置
- Vec3< T > m_gcell_max
  ガイドセルを含めた担当領域の最大位置
- BBox< T > m_gcell_bbox
  ガイドセルを含めた*Bounding Box*

6.2.1 Detailed Description

template<typename T>struct PolylibNS::CalcAreaInfo< T >

クラス:CalcAreaInfo 計算領域情報。

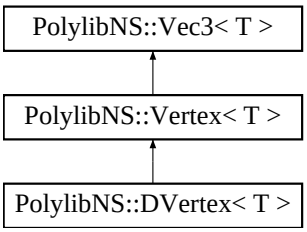
The documentation for this struct was generated from the following file:

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/Polylib.h

6.3 PolylibNS::DVertex< T > Class Template Reference

`#include <DVertex.h>`

Inheritance diagram for PolylibNS::DVertex< T >:



Public Member Functions

- **DVertex** (DVertexManager *DVM)
- void **set_DVM** (DVertexManager *DVM)
- void set_scalar (const int i, const T d)
- T get_scalar (int i) const
- void set_vector (const int i, const Vec3< T > vec)
- void get_vector (const int i, Vec3< T > *vec)
- DVertexManager * **DVM** ()

Additional Inherited Members

6.3.1 Detailed Description

template<typename T>class PolylibNS::DVertex< T >

クラス:DVertex polygon の頂点クラス。 Vertex クラスを継承　任意の実数型のスカラーデータを保持できる。

6.3.2 Member Function Documentation

6.3.2.1 template<typename T> T PolylibNS::DVertex< T >::get_scalar (int *i*) const [inline]

スカラー値の参照

Parameters

in	*i*	スカラーのインデックス。0 で開始。

Returns

スカラー値

6.3.2.2 template<typename T> void PolylibNS::DVertex< T >::get_vector (const int *i*, Vec3< T > * *vec*) [inline]

ベクター値の参照

Parameters

in	*i*	i 番目ベクトルのインデックス。0 で開始。
in	*vec*	Vec3 ベクトル

6.3.2.3 template<typename T> void PolylibNS::DVertex< T >::set_scalar (const int *i*, const T *d*) [inline]

スカラー値の登録 スカラー値を登録する。

Parameters

in	*i*	スカラーのインデックス。0 で開始。
in	*d*	登録するスカラー値。

6.3.2.4 template<typename T> void PolylibNS::DVertex< T >::set_vector (const int *i*, const Vec3< T > *vec*) [inline]

ベクター値の登録 ベクター値を登録する。

Parameters

in	*i*	i 番目ベクトルのインデックス。0 で開始。
in	*vec*	登録する Vec3<T> ベクター

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/DVertex.h

6.4 PolylibNS::DVertexManager Class Reference

`#include <DVertexManager.h>`

Public Member Functions

- **DVertexManager** (int nscalar, int nvector)
- int **nscalar** ()
- int **nvector** ()

6.4.1 Detailed Description

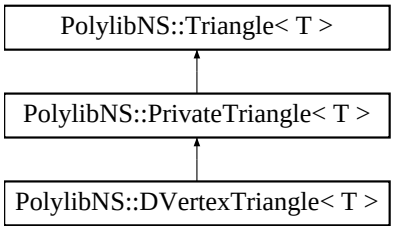
クラス:DVertexManager DVertex のユーザー定義データ形式を保持します。

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/DVertexManager.h

6.5 PolylibNS::DVertexTriangle< T > Class Template Reference

Inheritance diagram for PolylibNS::DVertexTriangle< T >:



Public Member Functions

- DVertexTriangle (DVertex< T > *vertex_ptr[3], int id)
- DVertexTriangle (DVertex< T > *vertex_ptr[3], Vec3< T > normal, int id)
- DVertexTriangle (DVertex< T > *vertex_ptr[3], Vec3< T > normal, T area, int id)
- DVertexTriangle (PrivateTriangle< T > tri, int id)
- DVertexTriangle (const DVertexTriangle< T > &tri)
- DVertexTriangle (T *dim, int id)
- DVertex< T > ** **get_DVertex** ()

Additional Inherited Members

6.5.1 Constructor & Destructor Documentation

6.5.1.1 template<typename T> PolylibNS::DVertexTriangle< T >::DVertexTriangle (DVertex< T > * *vertex_ptr[3]*, int *id*) [inline]

コンストラクタ。

Parameters

in	*vertex_ptr*	ポリゴンの頂点 (DVertex) へのポインタ。
in	*id*	三角形ポリゴンID。

6.5.1.2 template<typename T> PolylibNS::DVertexTriangle< T >::DVertexTriangle (DVertex< T > * *vertex_ptr[3]*, Vec3< T > *normal*, int *id*) [inline]

コンストラクタ。

Parameters

in	*vertex_ptr*	ポリゴンの頂点へのポインタ。
in	*normal*	法線。
in	*id*	三角形ポリゴンID。

6.5.1.3 template<typename T> PolylibNS::DVertexTriangle< T >::DVertexTriangle (DVertex< T > * *vertex_ptr[3]*, Vec3< T > *normal*, T *area*, int *id*) [inline]

コンストラクタ。

Parameters

in	*vertex_ptr*	ポリゴンの頂点へのポインタ。
in	*normal*	法線。
in	*area*	ポリゴンの面積。
in	*id*	三角形ポリゴンID。

6.5.1.4 template<typename T> PolylibNS::DVertexTriangle< T >::DVertexTriangle (PrivateTriangle< T > *tri*, int *id*) [inline]

コンストラクタ。

Parameters

in	*tri*	ポリゴン。
in	*id*	三角形ポリゴンID。

6.5.1.5 template<typename T> PolylibNS::DVertexTriangle< T >::DVertexTriangle (const DVertexTriangle< T > & *tri*) [inline]

コンストラクタ。

Parameters

in	*tri*	ポリゴン。

6.5.1.6 template<typename T> PolylibNS::DVertexTriangle< T >::DVertexTriangle (T * *dim*, int *id*) [inline]

コンストラクタ。

Parameters

in	*dim*	ポリゴン頂点座標配列。
in	*id*	三角形ポリゴンID。

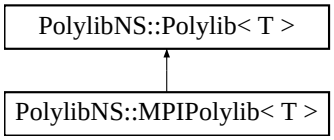
The documentation for this class was generated from the following file:

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/DVertexTriangle.h

6.6 PolylibNS::MPIPolylib< T > Class Template Reference

`#include <MPIPolylib.h>`

Inheritance diagram for PolylibNS::MPIPolylib< T >:



Public Member Functions

- POLYLIB_STAT init_parallel_info (MPI_Comm comm, T bpos[3], unsigned int bbsize[3], unsigned int gcsize[3], T dx[3])
- POLYLIB_STAT load (std::string config_filename)
- POLYLIB_STAT load_rank0 (std::string config_filename="", T scale=1.0)
- POLYLIB_STAT load_parallel (std::string config_filename="", ID_FORMAT id_format=ID_BIN)
- POLYLIB_STAT save (std::string *p_config_filename)
- POLYLIB_STAT save_rank0 (std::string *p_config_filename, std::string stl_format, std::string extend="")
- POLYLIB_STAT save_parallel (std::string *p_config_filename, std::string stl_format, std::string extend="", ID_FORMAT id_format=ID_BIN)
- POLYLIB_STAT move (PolylibMoveParams &params)
- POLYLIB_STAT migrate ()
- ParallelInfo< T > get_myproc ()
- unsigned int used_memory_size ()

Static Public Member Functions

- static MPIPolylib< T > * get_instance ()

Protected Member Functions

- MPIPolylib ()
- ~MPIPolylib ()
- void show_group_name (PolygonGroup< T > *p, std::string tab)
- POLYLIB_STAT broadcast_config (std::string config_contents)
- POLYLIB_STAT send_polygons_to_all ()
- POLYLIB_STAT pack_num_trias (std::vector< int > *p_vec, int group_id, const std::vector< PrivateTriangle< T > * > *p_trias)
- POLYLIB_STAT pack_trias (std::vector< T > *p_vec, const std::vector< PrivateTriangle< T > * > *p_trias)
- POLYLIB_STAT pack_tria_ndata (std::vector< int > *p_vec, const std::vector< PrivateTriangle< T > * > *p_trias)
- POLYLIB_STAT pack_tria_scalar_data (std::vector< T > *p_vec, const std::vector< PrivateTriangle< T > * > *p_trias)
- POLYLIB_STAT pack_tria_vector_data (std::vector< T > *p_vec, const std::vector< PrivateTriangle< T > * > *p_trias)
- POLYLIB_STAT pack_tria_ids (std::vector< int > *p_vec, const std::vector< PrivateTriangle< T > * > *p_trias)
- POLYLIB_STAT erase_outbounded_polygons ()
- POLYLIB_STAT broadcast_config_from_rank0 ()
- POLYLIB_STAT receive_polygons_from_rank0 ()
- POLYLIB_STAT gather_polygons ()
- POLYLIB_STAT send_polygons_to_rank0 ()
- POLYLIB_STAT gather_polygons_vtk ()
- POLYLIB_STAT send_polygons_to_rank0_vtk ()
- POLYLIB_STAT select_excluded_trias (PolygonGroup< T > *p_pg)
- ParallelInfo< T > * get_proc (int rank)

Protected Attributes

- ParallelInfo< T > m_myproc
  自*PE* 担当領域情報
- std::vector< ParallelInfo< T > * > m_other_procs
  自*PE* を除く全*PE* 担当領域情報リスト
- std::vector< ParallelInfo< T > * > m_neibour_procs
  隣接*PE* 担当領域情報リスト
- int m_myrank
  自プロセスのランク数
- int m_numproc
  全プロセス数
- MPI_Comm m_mycomm
  自プロセスが利用するコミュニケーター

6.6.1 Detailed Description

template<typename T>class PolylibNS::MPIPolylib< T >

クラス:MPIPolylib ポリゴンを管理する為の並列版クラスライブラリです。

6.6.2 Constructor & Destructor Documentation

6.6.2.1 template<typename T > PolylibNS::MPIPolylib< T >::MPIPolylib ( ) [protected]

コンストラクタ。 singleton のため非公開。本クラスインスタンス取得には get_instance() を利用する。

6.6.2.2 template<typename T > PolylibNS::MPIPolylib< T >::~MPIPolylib ( ) [protected]

デストラクタ。

6.6.3 Member Function Documentation

6.6.3.1 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::broadcast_config (std::string *config_contents*) [protected]

設定ファイル内容を他 rank へ broadcast する。

Parameters

in	*config_contents*	初期化ファイル内容。

Returns

POLYLIB_STAT で定義される値が返る。

6.6.3.2 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::broadcast_config_from_rank0 ( ) [protected]

ポリゴングループ定義情報を rank0 から受信し、グループ階層構造を構築。

Returns

POLYLIB_STAT で定義される値が返る。

6.6.3.3 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::erase_outbounded_polygons ( ) [protected]

自領域内ポリゴンのみ抽出してポリゴン情報を再構築。 migrate 実行後に行う。

Returns

POLYLIB_STAT で定義される値が返る。

6.6.3.4 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::gather_polygons ( ) [protected]

他 rank からポリゴン情報を rank0 で受信

6.6.3.5 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::gather_polygons_vtk ( ) [protected]

他 rank からポリゴン情報を rank0 で受信 (vtk)

6.6.3.6 template<typename T > MPIPolylib< T > * PolylibNS::MPIPolylib< T >::get_instance ( ) [static]

インスタンス取得。本クラスは singlton クラスです。

Returns

MPIPolylib クラスのインスタンス

6.6.3.7 template<typename T> ParallelInfo<T> PolylibNS::MPIPolylib< T >::get_myproc ( ) [inline]

m_myproc の内容を get

Returns

自PE 領域情報

6.6.3.8 template<typename T > ParallelInfo< T > * PolylibNS::MPIPolylib< T >::get_proc (int *rank*) [protected]

プロセス担当領域クラスのポインタを返す

Parameters

in	*rank*	ランク数

Returns

プロセス担当領域クラスのポインタ

6.6.3.9 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::init_parallel_info (MPI_Comm *comm*, T *bpos[3]*, unsigned int *bbsize[3]*, unsigned int *gcsize[3]*, T *dx[3]*)

並列計算関連情報の設定と初期化を行う。 全 rank で各々設定を行い、その領域情報を全 rank へ配信する。

Parameters

in	*comm*	MPI コミュニケーター
in	*bpos*	自PE 担当領域の基点座標
in	*bbsize*	同、計算領域のボクセル数
in	*gcsize*	同、ガイドセルのボクセル数
in	*dx*	同、ボクセル 1 辺の長さ

Returns

POLYLIB_STAT で定義される値が返る。

6.6.3.10 template<typename T> POLYLIB_STAT PolylibNS::MPIPolylib< T >::load (std::string *config_filename*) [inline]

Polylib::load()のオーバライドメソッド。

Attention

並列環境では利用できません。

Parameters

in	*config_filename*	初期化ファイル名。

Returns

常に PLSTAT_NG が返ります。

6.6.3.11 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::load_parallel (std::string *config_filename* = "", ID_FORMAT *id_format* = ID_BIN)

全 rank 並列でのデータ構築。 指定された設定ファイルを各 rank にて読み込み、グループ階層構造の構築、およびポリゴンデータの構築を行う。

Attention

各 rank が読み込むファイルに記述されたグループ階層構造が一致している必要がある。

Parameters

in	*config_filename*	初期化ファイル名。未指定時はデフォルトファイルを読む。
in	*id_format*	三角形ID ファイルの入力形式。

Returns

POLYLIB_STAT で定義される値が返る。

6.6.3.12 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::load_rank0 (std::string *config_filename* = "", T *scale* = 1.0)

rank0 によるデータ構築。 指定された設定ファイルを rank0 にて読み込み、グループ階層構造の構築 およびポリゴンデータの構築を行う。 グループ階層構造は全 rank に b_cast され、情報を共有する。 ポリゴンデータは各 rank 領域毎のデータが分配される。

Parameters

in	*config_filename*	初期化ファイル名。未指定時はデフォルトファイルを読む。

Returns

POLYLIB_STAT で定義される値が返る。

6.6.3.13 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::migrate ( )

ポリゴンデータのPE 間移動。 本クラスインスタンス配下の全PolygonGroup のポリゴンデータについて、move メソッドにより移動した三角形ポリゴン情報を隣接PE 間でやり取りする。

Returns

POLYLIB_STAT で定義される値が返る。

6.6.3.14 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::move (PolylibMoveParams & *params*)

ポリゴン座標の移動。 本クラスインスタンス配下の全PolygonGroup の move メソッドが呼び出される。

Parameters

in	*params*	移動計算要パラメタセット。

Returns

POLYLIB_STAT で定義される値が返る。

6.6.3.15 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::pack_num_trias (std::vector< int > * *p_vec*, int *group_id*, const std::vector< PrivateTriangle< T > * > * *p_trias*) [protected]

グループID &グループ内三角形数の送信情報を作成。

Parameters

in,out	*p_vec*	情報追加先ベクタ
in	*group_id*	グループID
in	*p_trias*	グループ内三角形リスト

Returns

POLYLIB_STAT で定義される値が返る。

6.6.3.16 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::pack_tria_ids (std::vector< int > * *p_vec*, const std::vector< PrivateTriangle< T > * > * *p_trias*) [protected]

三角形ID の送信情報を作成。

Parameters

in,out	*p_vec*	情報追加先ベクタ
in	*p_trias*	グループ内三角形リスト

Returns

POLYLIB_STAT で定義される値が返る。

6.6.3.17 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::pack_tria_ndata (std::vector< int > * *p_vec*, const std::vector< PrivateTriangle< T > * > * *p_trias*) [protected]

DVertex 三角形のデータ数を作成

Parameters

in,out	*p_vec*	情報追加先ベクタ
in	*p_trias*	グループ内三角形リスト

Returns

POLYLIB_STAT で定義される値が返る。

6.6.3.18 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::pack_tria_scalar_data (std::vector< T > * *p_vec*, const std::vector< PrivateTriangle< T > * > * *p_trias*) [protected]

DVertex 三角形のスカラーデータ送信情報を作成。

Parameters

in,out	*p_vec*	情報追加先ベクタ
in	*p_trias*	グループ内三角形リスト

Returns

POLYLIB_STAT で定義される値が返る。

6.6.3.19 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::pack_tria_vector_data (std::vector< T > * *p_vec*, const std::vector< PrivateTriangle< T > * > * *p_trias*) [protected]

DVertex 三角形のベクターデータ送信情報を作成。

Parameters

in,out	*p_vec*	情報追加先ベクタ
in	*p_trias*	グループ内三角形リスト

Returns

POLYLIB_STAT で定義される値が返る。

6.6.3.20 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::pack_trias (std::vector< T > * *p_vec*, const std::vector< PrivateTriangle< T > * > * *p_trias*) [protected]

三角形の送信情報を作成。

Parameters

in,out	*p_vec*	情報追加先ベクタ
in	*p_trias*	グループ内三角形リスト

Returns

POLYLIB_STAT で定義される値が返る。

6.6.3.21 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::receive_polygons_from_rank0 ( ) [protected]

自領域に必要なポリゴン情報を rank0 から受信

Returns

POLYLIB_STAT で定義される値が返る。

6.6.3.22 template<typename T> POLYLIB_STAT PolylibNS::MPIPolylib< T >::save (std::string * *p_config_filename*) [inline]

Polylib::save()のオーバライドメソッド。

Attention

並列環境では利用できません。

Parameters

out	*p_config_filename*	初期化ファイル名。

Returns

常に PLSTAT_NG が返ります。

6.6.3.23 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::save_parallel (std::string * *p_config_filename*, std::string *stl_format*, std::string *extend* = " ", ID_FORMAT *id_format* = ID_BIN)

全 rank 並列でのデータ保存。 各 rank の本クラスインスタンスが保持するグループ階層構造を設定ファイルに各 rank 毎に書き出す。 同時にポリゴンデータも指定されたフォーマットのSTL/OBJ ファイルに各 rank 毎に書き出す。 設定ファイル命名規則は以下の通り polylib_config_ランク番号_付加文字列.tpp STL/OBJ ファイル命名規則は以下の通り ポリゴングループ名称_ランク番号_付加文字列.拡張子

Parameters

out	*p_config_filename*	設定ファイル名返却用 string インスタンスへのポインタ
in	*stl_format*	STL/OBJ ファイルフォーマット。 "stl_a":アスキー形式　"stl_b":バイナリ形式 "obj_a":アスキー形式　"obj_b","obj_bb":バイナリ形式,"obj_bb"は、頂点法線付き。
in	*extend*	ファイル名に付加する文字列。省略可。省略 した場合は、付加文字列として本メソッド呼 び出し時の年月日時分秒 (YYYYMMDD24hhmmss) を用いる。
in	*id_format*	三角形ID ファイルの出力形式。

Returns

POLYLIB_STAT で定義される値が返る。

6.6.3.24 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::save_rank0 (std::string * *p_config_filename*, std::string *stl_format*, std::string *extend* = " ")

rank0 によるデータ保存。 rank0 の本クラスインスタンスが保持するグループ階層構造を設定ファイルに書き出す。 同時に各 rank に分散するポリゴンデータも rank0 に集められ、指定されたフォーマットの STL/OBJ ファイルに rank0 で書き出す。 設定ファイル命名規則は以下の通り polylib_config_付加文字列.tpp STL/OBJ ファイル命名規則は以下の通り ポリゴングループ名称_付加文字列.拡張子

Parameters

out	*p_config_filename*	設定ファイル名返却用 string インスタンスへのポインタ
in	*stl_format*	STL/OBJ ファイルフォーマット。 "stl_a":アスキー形式　"stl_b":バイナリ形式 "obj_a":アスキー形式　"obj_b","obj_bb":バイナリ形式,"obj_bb"は、頂点法線付き。
in	*extend*	ファイル名に付加する文字列。省略可。省略 した場合は、付加文字列として本メソッド呼 び出し時の年月日時分秒 (YYYYMMDD24hhmmss) を用いる。

Returns

POLYLIB_STAT で定義される値が返る。

Attention

出力引数 p_config_filename の返却値は rank0 でのみ有効

6.6.3.25 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::select_excluded_trias (PolygonGroup< T > * *p_pg*) [protected]

移動除外三角形ID リストの作成

6.6.3.26 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::send_polygons_to_all ( ) [protected]

各PE 領域内ポリゴン情報を全 rank に送信

Returns

POLYLIB_STAT で定義される値が返る。

6.6.3.27 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::send_polygons_to_rank0 ( ) [protected]

rank0 へポリゴン情報を送信

6.6.3.28 template<typename T > POLYLIB_STAT PolylibNS::MPIPolylib< T >::send_polygons_to_rank0_vtk ( ) [protected]

rank0 へポリゴン情報を送信 (vtk)

6.6.3.29 template<typename T> void PolylibNS::MPIPolylib< T >::show_group_name (PolygonGroup< T > * *p*, std::string *tab*) [protected]

指定されたグループ以下の階層構造をツリー形式で標準出力に出力する。

Parameters

p	表示対象となるグループのポインタ。
tab	階層の深さを示すスペース。

Attention

プロセス毎に動作する。 出力にランク数が加わる以外は非並列版と同じ。

6.6.3.30 template<typename T > unsigned int PolylibNS::MPIPolylib< T >::used_memory_size ( )

MPIPolylib が利用中の概算メモリ量を返す

Returns

利用中のメモリ量 (byte)

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/MPIPolylib.h

6.7 PolylibNS::ParallelInfo< T > Struct Template Reference

`#include <MPIPolylib.h>`

Public Attributes

- MPI_Comm m_comm
  MPI **コミュニケータ**
- int m_rank
  ランク数
- CalcAreaInfo< T > m_area
  計算領域情報
- std::map< int, std::vector< int > > m_exclusion_map
  migrate **除外三角形***ID* **マップ** *(k:***グループ***ID, v:***三角形***ID* **リスト***)*

6.7.1 Detailed Description

template<typename T>struct PolylibNS::ParallelInfo< T >

クラス:ParallelInfo 並列プロセス情報。

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/MPIPolylib.h

6.8 PolylibNS::PolygonGroup< T > Class Template Reference

`#include <PolygonGroup.h>`

Public Member Functions

- PolygonGroup ()
- PolygonGroup (T tolerance)
- virtual ~PolygonGroup ()
- POLYLIB_STAT init (const std::vector< PrivateTriangle< T > * > *tri_list, bool clear=true)
- POLYLIB_STAT init (const T *vertlist, const int *idlist, const int n_start_tri, const int n_start_id, const unsigned int n_tri)
- POLYLIB_STAT init_dvertex (const T *vertlist, const int *idlist, const T *scalarlist, const T *vectorlist, const int n_start_tri, const int n_start_id, const int n_start_scalar, const int n_start_vector, const unsigned int n_tri, const int n_scalar, const int n_vector)
- POLYLIB_STAT add_dvertex (const T *vertlist, const int *idlist, const T *scalarlist, const T *vectorlist, const int n_start_tri, const int n_start_id, const int n_start_scalar, const int n_start_vector, const unsigned int n_tri, const int n_scalar, const int n_vector)
- virtual POLYLIB_STAT build_group_tree (Polylib< T > *polylib, PolygonGroup< T > *parent, TextParser *tp)
- virtual POLYLIB_STAT build_group_tree (Polylib< T > *polylib, PolygonGroup< T > *parent, std::string path)
- POLYLIB_STAT replace_DVertex (int nscalar, int nvector)
- POLYLIB_STAT prepare_DVertex (int nscalar, int nvector)
- DVertexTriangle< T > * add_DVertex_Triangle (Vec3< T > *v)
- void finalize_DVertex ()
- POLYLIB_STAT build_polygon_tree ()

- POLYLIB_STAT load_stl_file (T scale=1.0)
- POLYLIB_STAT load_id_file (ID_FORMAT id_format)
- POLYLIB_STAT save_stl_file (std::string rank_no, std::string extend, std::string format, std::map< std::string, std::string > &stl_fname_map)
- POLYLIB_STAT save_id_file (std::string rank_no, std::string extend, ID_FORMAT id_format)
- virtual POLYLIB_STAT mk_param_tag (TextParser *pt, std::string rank_no, std::string extend, std::string format)
- virtual POLYLIB_STAT move (PolylibMoveParams &params)
- const std::vector < PrivateTriangle< T > * > * search (BBox< T > *bbox, bool every) const
- POLYLIB_STAT search (BBox< T > *bbox, bool every, std::vector< PrivateTriangle< T > * > *tri_list) const
- const std::vector < PrivateTriangle< T > * > * linear_search (BBox< T > *bbox, bool every) const
- POLYLIB_STAT linear_search (BBox< T > *bbox, bool every, std::vector< PrivateTriangle< T > * > *tri_list) const
- const PrivateTriangle< T > * search_nearest (const Vec3< T > &pos) const
- std::string acq_fullpath ()
- std::string acq_file_name ()
- const std::vector < PrivateTriangle< T > * > * search_outbounded (BBox< T > neibour_bbox, std::vector< int > *exclude_tria_ids)
- POLYLIB_STAT add_triangles (std::vector< PrivateTriangle< T > * > *tri_list)
- POLYLIB_STAT add_triangles (const T *vertlist, const int *idlist, const int n_start_tri, const int n_start_id, const unsigned int n_tri)
- POLYLIB_STAT rebuild_polygons ()
- POLYLIB_STAT show_group_info (int irank=-1)
- int get_group_num_tria (void)

  ポリゴングループの要素数を返す
- T get_group_area (void)

  ポリゴンの面積を積算して返す
- POLYLIB_STAT rescale_polygons (T scale)

  ポリゴンの縮尺変換&*KD* 木再構築
- POLYLIB_STAT set_all_exid_of_trias (int id)
- virtual std::string whoami ()
- void set_file_name (std::map< std::string, std::string > fname)
- std::map< std::string, std::string > get_file_name () const
- void set_name (std::string name)
- std::string get_name (void)
- void set_parent_path (std::string ppath)
- std::string get_parent_path (void)
- void set_parent (PolygonGroup< T > *p)
- PolygonGroup< T > * get_parent (void)
- void set_children (std::vector< PolygonGroup< T > * > &p)
- std::vector< PolygonGroup< T > * > & get_children (void)
- void add_children (PolygonGroup< T > *p)
- VertexList< T > * get_vertexlist ()
- VertKDT< T > * get_vertkdt ()
- std::vector< PrivateTriangle < T > * > * get_triangles ()
- VTree< T > * get_vtree ()
- int get_internal_id ()
- std::string get_label ()
- std::string get_type ()

- int get_id ()
- int get_movable ()
- size_t get_num_of_trias_before_move ()
- void print_vertex () const
- void **show_bbox** () const

Static Public Member Functions

- static std::string get_class_name ()

Static Public Attributes

- static const char * ATT_NAME_CLASS = "class_name"
- static const char * ATT_NAME_TOLERANCE = "tolerance"

Protected Member Functions

- POLYLIB_STAT setup_attribute (Polylib< T > *polylib, PolygonGroup< T > *parent, TextParser *tp)
- POLYLIB_STAT init_check_leaped ()
- POLYLIB_STAT check_leaped (Vec3< T > origin, Vec3< T > cell_size)
- bool is_far (Vec3< T > origin, Vec3< T > cell_size, Vec3< T > pos1, Vec3< T > pos2)

Protected Attributes

- int m_internal_id
  グループ*ID*。
- std::string m_name
  自グループ名。
- std::string m_parent_path
  親グループのパス名。
- PolygonGroup< T > * m_parent
  親グループへのポインタ。
- std::vector< PolygonGroup< T > * > m_children
  子グループへのポインタリスト。
- std::map< std::string, std::string > m_file_name
  STL **ファイル名とファイル形式。**
- Polygons< T > * m_polygons
  三角形*Polygons* **クラス。**
- bool m_movable
  move **メソッドにより移動するグループか？**
- bool m_need_rebuild
  KD **木の再構築が必要か？**
- std::vector< PrivateTriangle < T > * > * m_trias_before_move
  *move()***による移動前三角形一時保存リスト。**
- std::string m_label
  ユーザ定義ラベル *: (***追加** *2012.08.31)*
- std::string m_type
  ユーザ定義タイプ *: (***追加** *2013.07.17)*

6.8.1 Detailed Description

template<typename T>class PolylibNS::PolygonGroup< T >

クラス:PolygonGroup ポリゴングループを管理するクラスです。

6.8.2 Constructor & Destructor Documentation

6.8.2.1 template<typename T > PolylibNS::PolygonGroup< T >::PolygonGroup ( )

コンストラクタ m_DVM_ptr=NULL;

6.8.2.2 template<typename T > PolylibNS::PolygonGroup< T >::PolygonGroup (T *tolerance*)

コンストラクタ

Parameters

in	*tolerance*	頂点同一性チェックの基準値（距離）

6.8.2.3 template<typename T > PolylibNS::PolygonGroup< T >::~PolygonGroup ( ) [virtual]

デストラクタ

6.8.3 Member Function Documentation

6.8.3.1 template<typename T > std::string PolylibNS::PolygonGroup< T >::acq_file_name ( )

カンマ区切りでSTL ファイル名リストを取得。

Returns

ファイル名リスト。

6.8.3.2 template<typename T > std::string PolylibNS::PolygonGroup< T >::acq_fullpath ( )

PolygonGroup のフルパス名を取得する。

Returns

フルパス名。

6.8.3.3 template<typename T> void PolylibNS::PolygonGroup< T >::add_children (PolygonGroup< T > * *p*) [inline]

子グループを追加。

Parameters

in	*p*	子グループ。

6.8.3.4 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::add_dvertex (const T * *vertlist,* const int * *idlist,* const T * *scalarlist,* const T * *vectorlist,* const int *n_start_tri,* const int *n_start_id,* const int *n_start_scalar,* const int *n_start_vector,* const unsigned int *n_tri,* const int *n_scalar,* const int *n_vector*)

引数で与えられる三角形ポリゴンリスト (DVertexTrianle) を作成する。

Parameters

in	*vertlist*	設定する三角形ポリゴン頂点リスト。
in	*idlist*	三角形の id。
in	*scalarlist*	設定するスカラーデータのリスト
in	*vectorlist*	設定するベクターデータのリスト
in	*n_start_tri*	vertlist の頂点開始位置
in	*n_start_id*	idlist の id 開始位置
in	*n_start_scalar*	scalarlist の開始位置
in	*n_start_vector*	vectorlist の開始位置
in	*n_tri*	加える三角形の数
in	*n_scalar*	頂点あたりのスカラーデータの数
in	*n_vector*	頂点あたりのベクターデータの数

Returns

POLYLIB_STAT で定義される値が返る。

Attention

TriMesh クラスの init() 参照。オーバーロードメソッドあり。

6.8.3.5 template<typename T > DVertexTriangle< T > * PolylibNS::PolygonGroup< T >::add_DVertex_Triangle (Vec3< T > * *v*)

DVertex 追加作成用

Parameters

in	*v*	頂点座標（3 点）

Returns

polygon への pointer

6.8.3.6 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::add_triangles (std::vector< PrivateTriangle< T > * > * *tri_list*)

三角形リストの追加。

Parameters

in	*tri_list*	三角形ポリゴンリストのポインタ。

Returns

POLYLIB_STAT で定義される値が返る。

Attention

三角形ID が重複した三角形は追加しない。KD 木の再構築はしない。

6.8.3.7 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::add_triangles (const T * *vertlist*, const int * *idlist*, const int *n_start_tri*, const int *n_start_id*, const unsigned int *n_tri*)

三角形リストの追加。

Parameters

in	*vertlist*	設定する三角形ポリゴン頂点リスト。
in	*idlist*	三角形の id。
in	*n_start_tri*	vertlist の頂点開始位置
in	*n_start_id*	idlist の id 開始位置
in	*n_tri*	加える三角形の数

Returns

POLYLIB_STAT で定義される値が返る。

Attention

TriMesh クラスの add() 参照。オーバーロードメソッドあり。

6.8.3.8 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::build_group_tree (Polylib< T > * *polylib,* PolygonGroup< T > * *parent,* TextParser * *tp*) [virtual]

PolygonGroup ツリーの作成。 設定ファイルの内容を再帰的に呼び出し、PolygonGroup ツリーを作成する。

Parameters

in	*polylib*	Polygon クラスのインスタンス
in	*parent*	親グループ
in	*tp*	TextParser のインスタンス

Returns

POLYLIB_STAT で定義される値が返る。

6.8.3.9 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::build_group_tree (Polylib< T > * *polylib,* PolygonGroup< T > * *parent,* std::string *path*) [virtual]

PolygonGroup ツリーの作成。DVertex 新規作成用 name で表されたパスにポリゴンツリーを再帰的に作成する。

Parameters

in	*polylib*	Polygon クラスのインスタンス
in	*parent*	親グループ
in	*path*	作成するパス。作成するごとに上位のパスを消して再帰する。

Returns

POLYLIB_STAT で定義される値が返る。

6.8.3.10 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::build_polygon_tree ( )

三角形ポリゴンの法線ベクトルの計算、面積の計算、KD 木の生成を行う。 三角形ポリゴンはTriMesh クラスが管理している。

Returns

POLYLIB_STAT で定義される値が返る。

Attention

TriMesh クラスの build() 参照。

6.8.3.11 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::check_leaped (Vec3< T > *origin*, Vec3< T > *cell_size*) [protected]

move()メソッド実行により、頂点が隣接セルよりも遠くへ移動した三角形情報 を報告（後処理）。該当する三角形について、以下の情報を cerr へ出力する。 ・ポリゴングループID ・三角形ID ・移動前/後の頂点座標

Parameters

in	*origin*	計算領域起点座標
in	*cell_size*	ボクセルサイズ

Returns

POLYLIB_STAT で定義される値が返る。

Attention

本メソッドはデバッグ用です。 派生クラスでオーバーライドした move() メソッド内で、座標移動 処理後に呼ぶこと。

6.8.3.12 template<typename T > void PolylibNS::PolygonGroup< T >::finalize_DVertex ( )

DVertex 追加作成後の重複頂点削除 KD 木の構築

6.8.3.13 template<typename T> std::vector<PolygonGroup<T>*>& PolylibNS::PolygonGroup< T >::get_children (void) [inline]

子グループを取得。

Returns

子グループのリスト。

6.8.3.14 template<typename T> static std::string PolylibNS::PolygonGroup< T >::get_class_name ( ) [inline], [static]

クラス名を取得。

Returns

クラス名。

Attention

本クラスを継承する場合、継承後のクラス名を返すように変更す ることる。

6.8.3.15 template<typename T> std::map<std::string, std::string> PolylibNS::PolygonGroup< T >::get_file_name ( ) const [inline]

STL ファイル名とファイルフォーマットの対応マップ取得。

Returns

STL ファイル名とファイルフォーマットの対応マップ。

6.8.3.16 template<typename T> int PolylibNS::PolygonGroup< T >::get_id ( ) [inline]

ユーザ定義ID を取得。 追加 2010.10.20

Returns

ユーザ定義ID。

6.8.3.17 template<typename T> int PolylibNS::PolygonGroup< T >::get_internal_id ( ) [inline]

ポリゴングループID を取得。 メンバー名修正 (m_id -> m_internal_id) 2010.10.20

Returns

ポリゴングループID。

6.8.3.18 template<typename T> std::string PolylibNS::PolygonGroup< T >::get_label ( ) [inline]

ユーザ定義ラベルを取得。 追加 2012.08.31

Returns

ユーザ定義ラベル。

6.8.3.19 template<typename T> int PolylibNS::PolygonGroup< T >::get_movable ( ) [inline]

移動対象フラグを取得。

Returns

移動対象フラグ。

6.8.3.20 template<typename T> std::string PolylibNS::PolygonGroup< T >::get_name (void) [inline]

グループ名を取得。

Returns

グループ名。

6.8.3.21 template<typename T> size_t PolylibNS::PolygonGroup< T >::get_num_of_trias_before_move ( ) [inline]

move()による移動前三角形一時保存リストの個数を取得。

Returns

一時保存リストサイズ。

6.8.3.22 template<typename T> PolygonGroup<T>* PolylibNS::PolygonGroup< T >::get_parent (void) [inline]

親グループを取得。

Returns

親グループのポインタ。

6.8.3.23 template<typename T> std::string PolylibNS::PolygonGroup< T >::get_parent_path (void) [inline]

親グループのフルパス名を取得。

Returns

親グループのフルパス名。

6.8.3.24 template<typename T> std::vector<PrivateTriangle<T>*>* PolylibNS::PolygonGroup< T >::get_triangles ( ) [inline]

Polygon クラスが管理する三角形ポリゴンリストを取得。

Returns

三角形ポリゴンリスト。

6.8.3.25 template<typename T> std::string PolylibNS::PolygonGroup< T >::get_type ( ) [inline]

ユーザ定義タイプを取得。 追加 2013.07.17

Returns

ユーザ定義タイプ。

6.8.3.26 template<typename T> VertexList<T>* PolylibNS::PolygonGroup< T >::get_vertexlist ( ) [inline]

Polygon クラスが管理する頂点リストを取得。

Returns

頂点リスト

6.8.3.27 template<typename T> VertKDT<T>* PolylibNS::PolygonGroup< T >::get_vertkdt ( ) [inline]

Polygon クラスが管理するKD 木クラスを取得。

Returns

KD 木ポリゴンリスト。

6.8.3.28 template<typename T> VTree<T>* PolylibNS::PolygonGroup< T >::get_vtree ( ) [inline]

Polygon クラスが管理するKD 木クラスを取得。

Returns

KD 木ポリゴンリスト。

6.8.3.29 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::init (const std::vector< PrivateTriangle< T > * > * *tri_list*, bool *clear* = true)

引数で与えられる三角形ポリゴンリストを複製し、KD 木の生成を行う。

Parameters

in	*tri_list*	設定する三角形ポリゴンリスト。
in	*clear*	true:ポリゴン複製、面積計算、KD 木生成を行う。 false:面積計算、KD 木生成だけを行う。

Returns

POLYLIB_STAT で定義される値が返る。

Attention

TriMesh クラスの init() 参照。オーバーロードメソッドあり。

6.8.3.30 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::init (const T * *vertlist*, const int * *idlist*, const int *n_start_tri*, const int *n_start_id*, const unsigned int *n_tri*)

引数で与えられる三角形ポリゴンリストを複製し、KD 木の生成を行う。

Parameters

in	*vertlist*	設定する三角形ポリゴン頂点リスト。
in	*idlist*	三角形の id。
in	*n_start_tri*	vertlist の頂点開始位置
in	*n_start_id*	idlist の id 開始位置
in	*n_tri*	加える三角形の数

Returns

POLYLIB_STAT で定義される値が返る。

Attention

TriMesh クラスの init() 参照。オーバーロードメソッドあり。

6.8.3.31 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::init_check_leaped ( ) [protected]

move()メソッド実行により、頂点が隣接セルよりも遠くへ移動した三角形情報 を報告（前処理）。

Returns

POLYLIB_STAT で定義される値が返る。

Attention

本メソッドはデバッグ用です。 派生クラスでオーバーライドした move() メソッド内で、座標移動 処理前に呼ぶこと。

6.8.3.32 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::init_dvertex (const T * *vertlist*, const int * *idlist*, const T * *scalarlist*, const T * *vectorlist*, const int *n_start_tri*, const int *n_start_id*, const int *n_start_scalar*, const int *n_start_vector*, const unsigned int *n_tri*, const int *n_scalar*, const int *n_vector*)

引数で与えられる三角形ポリゴンリスト (DVertexTrianle) を複製し、KD 木の生成を行う。

Parameters

in	*vertlist*	設定する三角形ポリゴン頂点リスト。
in	*idlist*	三角形の id。
in	*scalarlist*	設定するスカラーデータのリスト
in	*vectorlist*	設定するベクターデータのリスト
in	*n_start_tri*	vertlist の頂点開始位置
in	*n_start_id*	idlist の id 開始位置
in	*n_start_scalar*	scalarlist の開始位置
in	*n_start_vector*	vectorlist の開始位置
in	*n_tri*	加える三角形の数
in	*n_scalar*	頂点あたりのスカラーデータの数
in	*n_vector*	頂点あたりのベクターデータの数

Returns

POLYLIB_STAT で定義される値が返る。

Attention

TriMesh クラスの init() 参照。オーバーロードメソッドあり。

6.8.3.33 template<typename T > bool PolylibNS::PolygonGroup< T >::is_far (Vec3< T > *origin*, Vec3< T > *cell_size*, Vec3< T > *pos1*, Vec3< T > *pos2*) [protected]

2 点が隣接ボクセルよりも離れているか？

Parameters

in	*origin*	計算領域起点座標。
in	*cell_size*	ボクセルサイズ。
in	*pos1*	点 (1)。
in	*pos2*	点 (2)。

Returns

true:2 点が隣接ボクセルよりも離れている。

6.8.3.34 template<typename T > const std::vector< PrivateTriangle< T > * > * PolylibNS::PolygonGroup< T >::linear_search (BBox< T > * *bbox*, bool *every*) const

線形探索により、指定矩形領域に含まれるポリゴンを抽出する。

Parameters

in	*bbox*	矩形領域。
in	*every*	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。

Returns

抽出したポリゴンリストのポインタ。

Attention

オーバーロードメソッドあり。

6.8.3.35 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::linear_search (BBox< T > * *bbox*, bool *every*, std::vector< PrivateTriangle< T > * > * *tri_list*) const

線形探索により、指定矩形領域に含まれるポリゴンを抽出する。

Parameters

in	*bbox*	矩形領域。
in	*every*	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。
in,out	*tri_list*	抽出した三角形ポリゴンリストのポインタ。

Returns

POLYLIB_STAT で定義される値が返る。

Attention

オーバーロードメソッドあり。

6.8.3.36 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::load_id_file (ID_FORMAT *id_format*)

三角形ポリゴンID ファイルからポリゴンID を読み込み、m_internal_id に登録する。

Parameters

in	*id_format*	三角形ID ファイルの入力形式。

Returns

POLYLIB_STAT で定義される値が返る。

6.8.3.37 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::load_stl_file (T *scale* = 1.0)

STL ファイルからポリゴン情報を読み込み、TriMesh クラスに登録する。

Returns

POLYLIB_STAT で定義される値が返る。

Attention

TriMesh クラスの import() 参照。

6.8.3.38 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::mk_param_tag (TextParser * *pt*, std::string *rank_no*, std::string *extend*, std::string *format*) [virtual]

設定ファイルに出力するTextParser のリーフを編集する. デフォルトでは何もしない。 CarGroup.cxx の例を参照.

Parameters

in	*pointer*	to TextParser
in	*rank_no*	ファイル名に付加するランク番号。
in	*extend*	ファイル名に付加する自由文字列。
in	*format*	STL ファイルフォーマット。

Returns

POLYLIB_STAT で定義される値が返る。

Attention

do nothing by default

6.8.3.39 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::move (PolylibMoveParams & *params*) [virtual]

三角形ポリゴン移動メソッド。virtual 用の関数なので処理はない。

Parameters

in	*params*	Polylib.h で宣言しているパラメタセットクラス。

Returns

POLYLIB_STAT で定義される値が返る。

6.8.3.40 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::prepare_DVertex (int *nscalar,* int *nvector*)

DVertex 追加作成用

Parameters

in	*nscalar*	スカラーデータ数
in	*nvector*	ベクターデータ数

Returns

POLYLIB_STAT で定義される値が返る。

6.8.3.41 template<typename T> void PolylibNS::PolygonGroup< T >::print_vertex ( ) const [inline]

test function for Vertex test

6.8.3.42 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::rebuild_polygons ( )

ポリゴン情報を再構築する。(KD 木の再構築をおこなう)

Returns

POLYLIB_STAT で定義される値が返る。

6.8.3.43 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::replace_DVertex (int *nscalar,* int *nvector*)

DVertex 追加作成用

Parameters

in	*nscalar*	スカラーデータ数
in	*nvector*	ベクターデータ数

Returns

POLYLIB_STAT で定義される値が返る。

6.8.3.44 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::save_id_file (std::string *rank_no*, std::string *extend*, ID_FORMAT *id_format*)

三角形ポリゴンID ファイルにポリゴンID を出力する。ID ファイル名は、 階層化されたグループ名_ランク番号_自由文字列.id。

Parameters

in	*rank_no*	ファイル名に付加するランク番号。
in	*extend*	ファイル名に付加する自由文字列。
in	*id_format*	三角形ID ファイルの出力形式。

Returns

POLYLIB_STAT で定義される値が返る。

6.8.3.45 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::save_stl_file (std::string *rank_no*, std::string *extend*, std::string *format*, std::map< std::string, std::string > & *stl_fname_map*)

TriMesh クラスが管理しているポリゴン情報をSTL ファイルに出力する。 TextParser 対応版

Parameters

in	*rank_no*	ファイル名に付加するランク番号。
in	*extend*	ファイル名に付加する自由文字列。
in	*format*	STL ファイルフォーマット。
in,out	*stl_fname_map*	stl ファイル名とポリゴングループのパス

Returns

POLYLIB_STAT で定義される値が返る。

Attention

TriMeshIO クラスの save() 参照。オーバーロードメソッドあり。

6.8.3.46 template<typename T > const std::vector< PrivateTriangle< T > * > * PolylibNS::PolygonGroup< T >::search (BBox< T > * *bbox*, bool *every*) const

KD 木探索により、指定矩形領域に含まれるポリゴンを抽出する。

Parameters

in	*bbox*	矩形領域。
in	*every*	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。

Returns

抽出したポリゴンリストのポインタ。

Attention

オーバーロードメソッドあり。

6.8.3.47 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::search (BBox< T > * *bbox*, bool *every*, std::vector< PrivateTriangle< T > * > * *tri_list*) const

KD 木探索により、指定矩形領域に含まれるポリゴンを抽出する。

Parameters

in	*bbox*	矩形領域。
in	*every*	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。
in,out	*tri_list*	抽出した三角形ポリゴンリストのポインタ。

Returns

POLYLIB_STAT で定義される値が返る。

Attention

オーバーロードメソッドあり。

6.8.3.48 template<typename T > const PrivateTriangle< T > * PolylibNS::PolygonGroup< T >::search_nearest (const Vec3< T > & *pos*) const

KD 木探索により、指定位置に最も近いポリゴンを検索する。

Parameters

in	*pos*	指定位置

Returns

検索されたポリゴン

6.8.3.49 template<typename T > const std::vector< PrivateTriangle< T > * > * PolylibNS::PolygonGroup< T >::search_outbounded (BBox< T > *neibour_bbox*, std::vector< int > * *exclude_tria_ids*)

PE 領域間移動する三角形ポリゴンリストの取得。

Parameters

in	*neibour_bbox*	隣接PE 領域バウンディングボックス。
in	*exclude_tria_ids*	領域移動対象外三角形ID リスト。

Returns

検索結果三角形リスト。

6.8.3.50 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::set_all_exid_of_trias (int *id*)

グループ配下の全Triangle オブジェクトの m_exid を更新する。

Parameters

in	*id*	更新するID 番号。

Returns

ステータスコード。

6.8.3.51 template<typename T> void PolylibNS::PolygonGroup< T >::set_children (std::vector< PolygonGroup< T > * > & *p*) [inline]

子グループを設定。

Parameters

in	*p*	子グループのリスト。

6.8.3.52 template<typename T> void PolylibNS::PolygonGroup< T >::set_file_name (std::map< std::string, std::string > *fname*) [inline]

STL ファイル名とファイルフォーマットを設定。

Parameters

in	*fname*	STL ファイル名とファイルフォーマットの対応マップ。

6.8.3.53 template<typename T> void PolylibNS::PolygonGroup< T >::set_name (std::string *name*) [inline]

グループ名を設定。

Parameters

in	*name*	グループ名。

6.8.3.54 template<typename T> void PolylibNS::PolygonGroup< T >::set_parent (PolygonGroup< T > * *p*) [inline]

親グループを設定。

Parameters

in	*p*	親グループのポインタ。

6.8.3.55 template<typename T> void PolylibNS::PolygonGroup< T >::set_parent_path (std::string *ppath*) [inline]

親グループのフルパス名を設定。

Parameters

in	*ppath*	親グループのフルパス名。

6.8.3.56 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::setup_attribute (Polylib< T > * *polylib*, PolygonGroup< T > * *parent*, TextParser * *tp*) [protected]

設定ファイルから取得したPolygonGroup の情報をインスタンスにセットする。

"filepath" に関して、先に filepath が複数　(filepath[0]) が存在するかどうか をチェックして、複数ならばその処理を行い、filepath の処理は終了する。 複数でないことが分かったら、filepath が単体で存在するかをチェックして、 存在するならば、処理を行う。

Parameters

in	*polylib*	Polygon クラスのインスタンス。
in	*parent*	親グループ。
in	*tp*	TextParser クラスのインスタンス

Returns

POLYLIB_STAT で定義される値が返る。

6.8.3.57 template<typename T > POLYLIB_STAT PolylibNS::PolygonGroup< T >::show_group_info (int *irank* = -1)

グループ情報（ランク番号、親グループ名、自分のグループ名、ファイル名、 頂点数、各頂点のXYZ 座標値、法線ベクトルのXYZ 座標値、面積）を出力する。

Parameters

in	*irank*	ランク数。

Returns

POLYLIB_STAT で定義される値が返る。

6.8.3.58 template<typename T> virtual std::string PolylibNS::PolygonGroup< T >::whoami ( ) [inline], [virtual]

クラス名を取得。

Returns

クラス名。

Attention

継承するクラスのクラス名取得関数 get_class_name() を呼び出す。

6.8.4 Member Data Documentation

6.8.4.1 template<typename T> const char * PolylibNS::PolygonGroup< T >::ATT_NAME_CLASS = "class_name" [static]

config ファイルに記述するParam タグのクラス名 (value="...")。
他クラスでも使用するXML タグ

6.8.4.2 template<typename T> const char * PolylibNS::PolygonGroup< T >::ATT_NAME_TOLERANCE = "tolerance" [static]

頂点同一性の基準値　config ファイルに記述するParam タグ (value="...")。

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/groups/PolygonGroup.h

6.9 PolylibNS::PolygonGroupFactory< T > Class Template Reference

`#include <PolygonGroupFactory.h>`

Public Member Functions

- PolygonGroupFactory ()
- virtual ~PolygonGroupFactory ()
- virtual PolygonGroup< T > * create_instance (std::string class_name, T tolerance)

6.9.1 Detailed Description

template<typename T>class PolylibNS::PolygonGroupFactory< T >

クラス:PolygonGroupFactory

6.9.2 Constructor & Destructor Documentation

6.9.2.1 template<typename T> PolylibNS::PolygonGroupFactory< T >::PolygonGroupFactory ( ) [inline]

コンストラクタ。

6.9.2.2 template<typename T> virtual PolylibNS::PolygonGroupFactory< T >::~PolygonGroupFactory ( ) [inline], [virtual]

デストラクタ。

6.9.3 Member Function Documentation

6.9.3.1 template<typename T> virtual PolygonGroup<T>* PolylibNS::PolygonGroupFactory< T >::create_instance (std::string *class_name,* T *tolerance*) [inline], [virtual]

インスタンス作成。

Parameters

in	*class_name*	作成するクラス名。

Returns

作成に失敗した場合はNULL が返る。

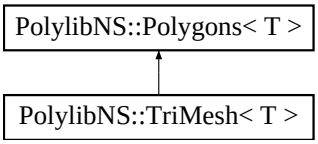
The documentation for this class was generated from the following file:

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/groups/PolygonGroupFactory.h

6.10 PolylibNS::Polygons< T > Class Template Reference

`#include <Polygons.h>`

Inheritance diagram for PolylibNS::Polygons< T >:



Public Member Functions

- Polygons ()
- virtual ~Polygons ()=0
- virtual void init (const std::vector< PrivateTriangle< T > * > *trias)=0
- virtual void init (const T *vertlist, const int *idlist, const int n_start_tri, const int n_start_id, const unsigned int n_tri)=0
- virtual void init_dvertex (const T *vertlist, const int *idlist, const T *scalarlist, const T *vectorlist, const int n_start_tri, const int n_start_id, const int n_start_scalar, const int n_start_vector, const unsigned int n_tri, const int n_scalar, const int n_vector)=0
- virtual void add (const std::vector< PrivateTriangle< T > * > *trias)=0
- virtual void add (const T *vertlist, const int *idlist, const int n_start_tri, const int n_start_id, const unsigned int n_tri)=0
- virtual void add_dvertex (const T *vertlist, const int *idlist, const T *scalarlist, const T *vectorlist, const int n_start_tri, const int n_start_id, const int n_start_scalar, const int n_start_vector, const unsigned int n_tri, const int n_scalar, const int n_vector)=0
- virtual POLYLIB_STAT import (const std::map< std::string, std::string > fname, T scale=1.0)=0
- virtual POLYLIB_STAT build ()=0
- virtual int triangles_num ()=0
- virtual const std::vector < PrivateTriangle< T > * > * search (BBox< T > *bbox, bool every) const =0
- virtual POLYLIB_STAT search (BBox< T > *bbox, bool every, std::vector< PrivateTriangle< T > * > *tri_list) const =0
- virtual const std::vector < PrivateTriangle< T > * > * linear_search (BBox< T > *bbox, bool every) const =0
- virtual POLYLIB_STAT linear_search (BBox< T > *bbox, bool every, std::vector< PrivateTriangle< T > * > *tri_list) const =0
- virtual const PrivateTriangle < T > * search_nearest (const Vec3< T > &pos) const =0
- virtual POLYLIB_STAT set_all_exid (const int id) const =0

- virtual POLYLIB_STAT replace_DVertex (int nscalar, int nvector)
- virtual POLYLIB_STAT prepare_DVertex (int nscalar, int nvector)
- virtual DVertexTriangle< T > * add_DVertex_Triangle (Vec3< T > *v)
- virtual void finalize_DVertex ()
- std::vector< PrivateTriangle < T > * > * get_tri_list () const
- VertexList< T > * get_vtx_list () const
- virtual VertKDT< T > * get_vertkdt () const =0
- virtual VTree< T > * get_vtree () const =0
- virtual void print_vertex ()
- virtual void **print_memory_size** () const =0
- virtual BBox< T > get_bbox () const =0
- virtual bool hasDVertex () const =0

Protected Attributes

- std::vector< PrivateTriangle < T > * > * m_tri_list
  三角形ポリゴンのリスト。
- VertexList< T > * **m_vertex_list**
- T **tolerance**

6.10.1 Detailed Description

template<typename T>class PolylibNS::Polygons< T >

クラス:Polygons 三角形ポリゴン集合を管理する純粋仮想クラスです。

6.10.2 Constructor & Destructor Documentation

6.10.2.1 template<typename T > PolylibNS::Polygons< T >::Polygons ( ) [inline]

コンストラクタ。

6.10.2.2 template<typename T > PolylibNS::Polygons< T >::~Polygons ( ) [pure virtual]

デストラクタ。
デストラクタ

Attention

継承しているクラスから呼び出されるために必要。

6.10.3 Member Function Documentation

6.10.3.1 template<typename T > virtual void PolylibNS::Polygons< T >::add (const std::vector< PrivateTriangle< T > * > * *trias*) [pure virtual]

三角形ポリゴンリストに引数で与えられる三角形を追加する。

Parameters

in	*trias*	設定する三角形ポリゴンリスト。

Implemented in PolylibNS::TriMesh< T >.

6.10.3.2 template<typename T > virtual void PolylibNS::Polygons< T >::add (const T * *vertlist*, const int * *idlist*, const int *n_start_tri*, const int *n_start_id*, const unsigned int *n_tri*) [pure virtual]

三角形ポリゴンリストに引数で与えられる三角形を追加する。

Parameters

in	*vertlist*	設定する三角形ポリゴン頂点リスト。
in	*idlist*	三角形の id。
in	*n_start_tri*	vertlist の頂点開始位置
in	*n_start_id*	idlist の id 開始位置
in	*n_tri*	加える三角形の数

Implemented in PolylibNS::TriMesh< T >.

6.10.3.3 template<typename T > virtual void PolylibNS::Polygons< T >::add_dvertex (const T * *vertlist*, const int * *idlist*, const T * *scalarlist*, const T * *vectorlist*, const int *n_start_tri*, const int *n_start_id*, const int *n_start_scalar*, const int *n_start_vector*, const unsigned int *n_tri*, const int *n_scalar*, const int *n_vector*) [pure virtual]

三角形ポリゴンリストに引数で与えられる三角形 (DVertexTriangle) を追加する。

Parameters

in	*vertlist*	設定する三角形ポリゴン頂点リスト。
in	*idlist*	三角形の id。
in	*scalarlist*	設定するスカラーデータのリスト
in	*vectorlist*	設定するベクターデータのリスト
in	*n_start_tri*	vertlist の頂点開始位置
in	*n_start_id*	idlist の id 開始位置
in	*n_start_scalar*	scalarlist の開始位置
in	*n_start_vector*	vectorlist の開始位置
in	*n_tri*	加える三角形の数
in	*n_scalar*	頂点あたりのスカラーデータの数
in	*n_vector*	頂点あたりのベクターデータの数

Implemented in PolylibNS::TriMesh< T >.

6.10.3.4 template<typename T > virtual DVertexTriangle<T>* PolylibNS::Polygons< T >::add_DVertex_Triangle (Vec3< T > * *v*) [inline], [virtual]

DVertex 追加作成用

Parameters

in	*v*	頂点座標（3 点）

Returns

polygon への pointer

Reimplemented in PolylibNS::TriMesh< T >.

6.10.3.5 template<typename T > virtual POLYLIB_STAT PolylibNS::Polygons< T >::build ( ) [pure virtual]

Polygons クラスに含まれる全ポリゴン情報からKD 木を作成する。

Returns

POLYLIB_STAT で定義される値が返る。

Implemented in PolylibNS::TriMesh< T >.

6.10.3.6 template<typename T > virtual void PolylibNS::Polygons< T >::finalize_DVertex ( ) [inline], [virtual]

DVertex 追加作成後の重複頂点削除

Reimplemented in PolylibNS::TriMesh< T >.

6.10.3.7 template<typename T > virtual BBox<T> PolylibNS::Polygons< T >::get_bbox ( ) const [pure virtual]

TriMesh クラスが管理しているBoundingBox を返す。

Implemented in PolylibNS::TriMesh< T >.

6.10.3.8 template<typename T > std::vector<PrivateTriangle<T>*>* PolylibNS::Polygons< T >::get_tri_list ( ) const [inline]

三角形ポリゴンのリストを取得。

Returns

三角形ポリゴンのリスト。

6.10.3.9 template<typename T > virtual VertKDT<T>* PolylibNS::Polygons< T >::get_vertkdt ( ) const [pure virtual]

VertexKD 木クラスを取得。

Returns

KD 木クラス。

Implemented in PolylibNS::TriMesh< T >.

6.10.3.10 template<typename T > virtual VTree<T>* PolylibNS::Polygons< T >::get_vtree ( ) const [pure virtual]

KD 木クラスを取得。

Returns

KD 木クラス。

Implemented in PolylibNS::TriMesh< T >.

6.10.3.11 template<typename T > VertexList<T>* PolylibNS::Polygons< T >::get_vtx_list ( ) const [inline]

VertexList を取得。

Returns

VertexList 頂点リストクラス

6.10.3.12 template<typename T > virtual bool PolylibNS::Polygons< T >::hasDVertex ( ) const [pure virtual]

hasDVertex

Returns

DVertex を持っているか

Implemented in PolylibNS::TriMesh< T >.

6.10.3.13 template<typename T > virtual POLYLIB_STAT PolylibNS::Polygons< T >::import (const std::map< std::string, std::string > *fname*, T *scale* = 1.0) [pure virtual]

STL ファイルを読み込みデータの初期化。

Parameters

in	*fname*	ファイル名とファイルフォーマットの map。

Returns

POLYLIB_STAT で定義される値が返る。

Implemented in PolylibNS::TriMesh< T >.

6.10.3.14 template<typename T > virtual void PolylibNS::Polygons< T >::init (const std::vector< PrivateTriangle< T > * > * *trias*) [pure virtual]

引数で与えられる三角形ポリゴンリストの複製を設定する。

Parameters

in	*trias*	設定する三角形ポリゴンリスト。

Attention

オーバーロードメソッドあり。

Implemented in PolylibNS::TriMesh< T >.

6.10.3.15 template<typename T > virtual void PolylibNS::Polygons< T >::init (const T * *vertlist*, const int * *idlist*, const int *n_start_tri*, const int *n_start_id*, const unsigned int *n_tri*) [pure virtual]

三角形ポリゴンリストを初期化し、引数で与えら れる三角形ポリゴンリストを設定する。 三角形ポリゴン用のメモリ領域は、TriMesh 内で新たに確保する。

Parameters

in	*vertlist*	設定する三角形ポリゴン頂点リスト。
in	*idlist*	三角形の id。
in	*n_start_tri*	vertlist の頂点開始位置
in	*n_start_id*	idlist の id 開始位置
in	*n_tri*	加える三角形の数

Implemented in PolylibNS::TriMesh< T >.

6.10.3.16 template<typename T > virtual void PolylibNS::Polygons< T >::init_dvertex (const T * *vertlist*, const int * *idlist*, const T * *scalarlist*, const T * *vectorlist*, const int *n_start_tri*, const int *n_start_id*, const int *n_start_scalar*, const int *n_start_vector*, const unsigned int *n_tri*, const int *n_scalar*, const int *n_vector*) [pure virtual]

三角形ポリゴンリストを初期化し、引数で与えら れる三角形ポリゴンリストを設定する。 三角形ポリゴン用のメモリ領域は、TriMesh 内で新たに確保する。

Parameters

in	*vertlist*	設定する三角形ポリゴン頂点リスト。
in	*idlist*	三角形の id。
in	*scalarlist*	設定するスカラーデータのリスト
in	*vectorlist*	設定するベクターデータのリスト
in	*n_start_tri*	vertlist の頂点開始位置
in	*n_start_id*	idlist の id 開始位置
in	*n_start_scalar*	scalarlist の開始位置
in	*n_start_vector*	vectorlist の開始位置
in	*n_tri*	加える三角形の数
in	*n_scalar*	頂点あたりのスカラーデータの数
in	*n_vector*	頂点あたりのベクターデータの数

Implemented in PolylibNS::TriMesh< T >.

6.10.3.17 template<typename T > virtual const std::vector<PrivateTriangle<T>*>* PolylibNS::Polygons< T >::linear_search (BBox< T > * *bbox*, bool *every*) const [pure virtual]

線形探索により、指定矩形領域に含まれる三角形ポリゴンを抽出する。

Parameters

in	*bbox*	検索範囲を示す矩形領域。
in	*every*	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。

Returns

抽出したポリゴンリストのポインタ。

Attention

MPIPolylib 内でのみ利用するため、ユーザは使用しないで下さい。
オーバーロードメソッドあり。

Implemented in PolylibNS::TriMesh< T >.

6.10.3.18 template<typename T > virtual POLYLIB_STAT PolylibNS::Polygons< T >::linear_search (BBox< T > * *bbox*, bool *every*, std::vector< PrivateTriangle< T > * > * *tri_list*) const [pure virtual]

線形探索により、指定矩形領域に含まれるポリゴンを抽出する。

Parameters

in	*bbox*	検索範囲を示す矩形領域。
in	*every*	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。
in,out	*tri_list*	抽出した三角形ポリゴンリストのポインタ。

Returns

POLYLIB_STAT で定義される値が返る。

Attention

オーバーロードメソッドあり。

Implemented in PolylibNS::TriMesh< T >.

6.10.3.19 template<typename T > virtual POLYLIB_STAT PolylibNS::Polygons< T >::prepare_DVertex (int *nscalar,* int *nvector*) [inline], [virtual]

Vertex -> DVertex への準備

Parameters

in	*nscalar*	スカラーデータ数
in	*nvector*	ベクトルデータ数

Reimplemented in PolylibNS::TriMesh< T >.

6.10.3.20 template<typename T > virtual void PolylibNS::Polygons< T >::print_vertex ( ) [inline], [virtual]

print_vertex test function for Vertex Class codes for VertexList

6.10.3.21 template<typename T > virtual POLYLIB_STAT PolylibNS::Polygons< T >::replace_DVertex (int *nscalar,* int *nvector*) [inline], [virtual]

Vertex -> DVertex へのリプレース

Parameters

in	*nscalar*	スカラーデータ数
in	*nvector*	ベクトルデータ数

Reimplemented in PolylibNS::TriMesh< T >.

6.10.3.22 template<typename T > virtual const std::vector<PrivateTriangle<T>*>* PolylibNS::Polygons< T >::search (BBox< T > * *bbox,* bool *every*) const [pure virtual]

KD 木探索により、指定矩形領域に含まれる三角形ポリゴンを抽出する。

Parameters

in	*bbox*	検索範囲を示す矩形領域。
in	*every*	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。

Returns

抽出したポリゴンリストのポインタ。

Attention

MPIPolylib 内でのみ利用するため、ユーザは使用しないで下さい。
オーバーロードメソッドあり。

Implemented in PolylibNS::TriMesh< T >.

6.10.3.23 template<typename T > virtual POLYLIB_STAT PolylibNS::Polygons< T >::search (BBox< T > * *bbox*, bool *every*, std::vector< PrivateTriangle< T > * > * *tri_list*) const [pure virtual]

KD 木探索により、指定矩形領域に含まれるポリゴンを抽出する。

Parameters

in	*bbox*	検索範囲を示す矩形領域。
in	*every*	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。
in,out	*tri_list*	抽出した三角形ポリゴンリストへのポインタ。

Returns

POLYLIB_STAT で定義される値が返る。

Attention

オーバーロードメソッドあり。

Implemented in PolylibNS::TriMesh< T >.

6.10.3.24 template<typename T > virtual const PrivateTriangle<T>* PolylibNS::Polygons< T >::search_nearest (const Vec3< T > & *pos*) const [pure virtual]

KD 木探索により、指定位置に最も近いポリゴンを検索する。

Parameters

in	*pos*	指定位置

Returns

検索されたポリゴン

Implemented in PolylibNS::TriMesh< T >.

6.10.3.25 template<typename T > virtual POLYLIB_STAT PolylibNS::Polygons< T >::set_all_exid (const int *id*) const [pure virtual]

配下の全ポリゴンの m_exid 値を指定値にする。

Parameters

in	*id*	指定値

Implemented in PolylibNS::TriMesh< T >.

6.10.3.26 template<typename T > virtual int PolylibNS::Polygons< T >::triangles_num ( ) [pure virtual]

Polygons クラスで保持する三角形ポリゴンの総数を返す。

Returns

三角形ポリゴンの総数。

Implemented in PolylibNS::TriMesh< T >.

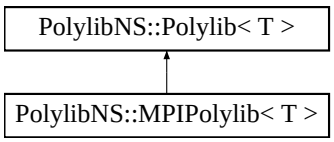
The documentation for this class was generated from the following files:

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/groups/PolygonGroup.h
- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/Polygons.h

6.11 PolylibNS::Polylib< T > Class Template Reference

`#include <Polylib.h>`

Inheritance diagram for PolylibNS::Polylib< T >:



Public Member Functions

- void set_factory (PolygonGroupFactory< T > *factory=NULL)
- POLYLIB_STAT load (std::string config_name="polylib_config.tpp", T scale=1.0)

  TextParser.
- POLYLIB_STAT save (std::string *p_config_name, std::string stl_format, std::string extend="")
- POLYLIB_STAT move (PolylibMoveParams &params)
- std::vector< PolygonGroup< T > * > * get_root_groups () const
- std::vector< PolygonGroup< T > * > * get_leaf_groups () const
- std::vector< Triangle< T > * > * search_polygons (std::string group_name, Vec3< T > min_pos, Vec3< T > max_pos, bool every) const
- const Triangle< T > * search_nearest_polygon (std::string group_name, const Vec3< T > &pos) const
- POLYLIB_STAT check_group_name (const std::string &pg_name, const std::string &parent_path)
- PolygonGroup< T > * create_polygon_group (std::string class_name, T tolerance)
- void add_pg_list (PolygonGroup< T > *pg)
- void show_group_hierarchy (FILE *fp=NULL)

  2010.10.20 **引数***FILE** **追加。*
- POLYLIB_STAT show_group_info (std::string group_name)
- unsigned int used_memory_size ()
- PolygonGroup< T > * get_group (std::string name) const
- std::string getVersionInfo ()

バージョン番号の文字列を返す

- void make_DVertex_PolygonGroup (std::string group_name, int nscalar, int nvector)
- DVertexTriangle< T > * add_DVertex_Triangle (std::string name, Vec3< T > *v)
- void finalize_DVertex (std::string name)

Static Public Member Functions

- static Polylib< T > * get_instance ()

Protected Member Functions

- Polylib ()
- ~Polylib ()
- POLYLIB_STAT make_group_tree (TextParser *tp_ptr)
- POLYLIB_STAT load_with_idfile (std::string config_name, ID_FORMAT id_format, T scale=1.0)

  textparser 版
- POLYLIB_STAT load_polygons (bool with_id_file, ID_FORMAT id_format, T scale=1.0)
- char * save_config_file (std::string rank_no, std::string extend, std::string format)
- POLYLIB_STAT clearfilepath (TextParser *tp_ptr)
- POLYLIB_STAT setfilepath (std::map< std::string, std::string > &stl_fname_map)
- void create_tp_structure (std::map< std::string, std::string > &stl_fname_map)
- char * polylib_config_save_file (std::string rank_no, std::string extend)

  設定ファイルの保存。 *PolylibConfig* 内部にあったものをここへ。
- POLYLIB_STAT save_with_rankno (std::string *p_config_name, int myrank, int maxrank, std::string extend, std::string stl_format, ID_FORMAT id_format)
- void show_group_name (PolygonGroup< T > *p, std::string tab, FILE *fp)
- PolygonGroup< T > * get_group (int internal_id) const

Protected Attributes

- PolygonGroupFactory< T > * m_factory

  自クラスのインスタンス *(singleton)*
- std::vector< PolygonGroup< T > * > m_pg_list

  ポリゴングループリスト
- TextParser * tp

  TextParser へのポインタ
- T m_distance_tolerance

  頂点を同一視する場合の基準値

6.11.1 Detailed Description

template<typename T>class PolylibNS::Polylib< T >

クラス:Polylib ポリゴンを管理する為のクラスライブラリです。

6.11.2 Constructor & Destructor Documentation

6.11.2.1 template<typename T > PolylibNS::Polylib< T >::Polylib ( ) [protected]

コンストラクタ

Attention

singleton のため、子クラス以外からの呼び出し不可とする

6.11.2.2 template<typename T > PolylibNS::Polylib< T >::~Polylib ( ) [protected]

デストラクタ

6.11.3 Member Function Documentation

6.11.3.1 template<typename T > DVertexTriangle< T > * PolylibNS::Polylib< T >::add_DVertex_Triangle (std::string *name*, Vec3< T > * *v*)

新規にDVertex を持つ三角形ポリゴンを追加する。

Parameters

in	*name*	ポリゴンを追加するポリゴングループの名称。
in	*v*	ポリゴンの各頂点　0〜2

Returns

作成したポリゴンへのポインタ。

6.11.3.2 template<typename T > void PolylibNS::Polylib< T >::add_pg_list (PolygonGroup< T > * *pg*)

PolygonGroup の追加。 本クラスが管理しているPolygonGroup のリストにPolygonGroup を追加する。

Parameters

in	*pg*	PolygonGroup

Attention

Polylib 内部で使用する関数であり、通常は利用者が用いるもの ではない。

6.11.3.3 template<typename T > POLYLIB_STAT PolylibNS::Polylib< T >::check_group_name (const std::string & *pg_name*, const std::string & *parent_path*)

引数のグループ名が既存グループと重複しないかチェック。

Parameters

in	*pg_name*	グループ名
in	*parent_path*	親グループまでのフルパス

Returns

POLYLIB_STAT で定義される値が返る。

Attention

Polylib 内部で使用する関数であり、通常は利用者が用いるもの ではない。

6.11.3.4 template<typename T > POLYLIB_STAT PolylibNS::Polylib< T >::clearfilepath (TextParser * *tp_ptr*) [protected]

TextParser 内部データから "filepath" "filepath[*]" というリーフを すべて削除する.

recursive の動作の為、引数に tp_ptr が必要

Parameters

in	*tp_ptr*	TextParser へのポインタ.

Returns

POLYLIB_STAT で定義される値が返る。

6.11.3.5 template<typename T > PolygonGroup< T > * PolylibNS::Polylib< T >::create_polygon_group (std::string *class_name,* T *tolerance*)

PolygonGroup のインスタンスの生成。 本クラスが管理しているFactory クラスを利用して、引数で渡されたクラス名 に応じたPolygonGroup のインスタンスを生成する。

Parameters

in	*class_name*	クラス名
in	**頂点同一性判定基準**	

Returns

生成したPolygonGroup

Attention

Polylib 内部で使用する関数であり、通常は利用者が用いるもの ではない。

6.11.3.6 template<typename T > void PolylibNS::Polylib< T >::create_tp_structure (std::map< std::string, std::string > & *stl_fname_map*) [protected]

map

Parameters

in	*stl_fname_map*	save したSTL ファイルとその階層の map 型データ

Returns

POLYLIB_STAT で定義される値が返る。

6.11.3.7 template<typename T > void PolylibNS::Polylib< T >::finalize_DVertex (std::string *name*)

新規にDVertex を登録した場合に、不要頂点の削除を行う。

Parameters

in	*name*	ポリゴングループ名称

6.11.3.8 template<typename T > PolygonGroup< T > * PolylibNS::Polylib< T >::get_group (std::string *name*) const

グループの取得。 name で与えられた名前のPolygonGroup を返す。

Parameters

in	*name*	グループ名

Returns

ポリゴングループクラスのポインタ。エラー時はNULL が返る。

Attention

オーバーロードメソッドあり。

6.11.3.9 template<typename T > PolygonGroup< T > * PolylibNS::Polylib< T >::get_group (int *internal_id*) const [protected]

グループの取得。 internal_id で与えられた m_internal_id を持つPolygonGroup を返す。

Parameters

in	*internal_id*	ポリゴングループID

Returns

ポリゴングループクラスのポインタ。エラー時はNULL が返る。

Attention

オーバーロードメソッドあり。

6.11.3.10 template<typename T > Polylib< T > * PolylibNS::Polylib< T >::get_instance ( ) [static]

singleton のPolylib インスタンス取得。 デフォルトのFactory クラスであるPolygonGroupFactory を使用してインスタンス を生成する。

Returns

Polylib クラスのインスタンス。

Attention

呼び出し側で delete はできません。

6.11.3.11 template<typename T > std::vector< PolygonGroup< T > * > * PolylibNS::Polylib< T >::get_leaf_groups ( ) const

リーフPolygonGroup リストの取得。 PolygonGroup ツリーの末端ノード（リーフ）をリスト化する。

Returns

リーフPolygonGroup の vector. 返却したPolygonGroup は削除不可。vector は要削除。

6.11.3.12 template<typename T > std::vector< PolygonGroup< T > * > * PolylibNS::Polylib< T >::get_root_groups ( ) const

PolygonGroup ツリーの最上位ノードの取得。

Returns

最上位ノードの vector。

Attention

返却したPolygonGroup は、削除不可。vector は要削除。

6.11.3.13 template<typename T > POLYLIB_STAT PolylibNS::Polylib< T >::load (std::string *config_name* = "polylib_config.tpp", T *scale* = 1.0)

TextParser.

PolygoGroup、三角形ポリゴン情報の読み込み。 引数で指定された設定ファイル (TextParser 形式) を読み込み、グループツリーを作成する。 続いて設定ファイルで指定されたSTL ファイルを読み込み、KD 木を作成する。

Parameters

in	*config_name*	設定ファイル名。

Returns

POLYLIB_STAT で定義される値が返る。

6.11.3.14 template<typename T > POLYLIB_STAT PolylibNS::Polylib< T >::load_polygons (bool *with_id_file*, ID_FORMAT *id_format*, T *scale* = 1.0) [protected]

STL ファイルの読み込み。 グループツリーの全リーフについて、設定されているSTL ファイルから ポリゴン情報を読み込む。読み込んだ後、KD 木の生成、法線の計算、面積の 計算を行う。

Parameters

in	*with_id_file*	true ならば、三角形ポリゴンID ファイルを読み 込んで m_id を設定する。 false ならば、STL 読み込み時に m_id を自動生成。
in	*id_format*	三角形ID ファイルの入力形式。

Returns

POLYLIB_STAT で定義される値が返る。

6.11.3.15 template<typename T > POLYLIB_STAT PolylibNS::Polylib< T >::load_with_idfile (std::string *config_name*, ID_FORMAT *id_format*, T *scale* = 1.0) [protected]

textparser 版

三角形ID ファイルの存在が必須な load 関数。 load と同様の動作を行う。但し読み込み時には、三角形ID ファイルが必要で あり、このファイルに記述されているID を用いて m_id を設定する。

Parameters

in	*config_name*	設定ファイル名。
in	*id_format*	三角形ID ファイルの入力形式。

Returns

POLYLIB_STAT で定義される値が返る。

Attention

MPIPolylib クラスがMPI 環境で利用することを想定している。

6.11.3.16 template<typename T > void PolylibNS::Polylib< T >::make_DVertex_PolygonGroup (std::string *group_name*, int *nscalar*, int *nvector*)

与えられた名前のポリゴングループにDVertex クラスを持つポリゴンを格納する 際の、スカラーデータとベクトルデータの数を指定する。 指定された名前のポリゴングループが存在しない場合は、 指定された階層にポリゴングループを新規に作成する。 DT は、データの型である。

Parameters

in	*group_name*	ポリゴングループの名称
in	*nscalar*	スカラーデータ数
in	*nvector*	ベクトルデータ数

6.11.3.17 template<typename T > POLYLIB_STAT PolylibNS::Polylib< T >::make_group_tree (TextParser * *tp_ptr*) [protected]

グループツリー作成。 TextParser クラスを使い、 PolygonGroup を作成し、グループツリーに登録する。

Parameters

in	*TextParser*	のインスタンス

Returns

POLYLIB_STAT で定義される値が返る。

Attention

オーバーロードメソッドあり。

6.11.3.18 template<typename T > POLYLIB_STAT PolylibNS::Polylib< T >::move (PolylibMoveParams & *params*)

三角形ポリゴン座標の移動。 本クラスインスタンス配下の全PolygonGroup の move メソッドが呼び出される。 move メソッドは、PolygonGroup クラスを拡張したクラスに利用者が記述する。

Parameters

in	*params*	Polylib.h で宣言された移動計算パラメータセット。

Returns

POLYLIB_STAT で定義される値が返る。

6.11.3.19 template<typename T > char * PolylibNS::Polylib< T >::polylib_config_save_file (std::string *rank_no*, std::string *extend*) [protected]

設定ファイルの保存。 PolylibConfig 内部にあったものをここへ。

Parameters

in	*rank_no*	ランク番号
in	*extend*	ファイル名に付加する文字列
in	*format*	TriMeshIO クラスで定義されているSTL ファイルのフォー マット。

Returns

作成した設定ファイルの名称。エラー時はNULL が返る。

6.11.3.20 template<typename T > POLYLIB_STAT PolylibNS::Polylib< T >::save (std::string * *p_config_name*, std::string *stl_format*, std::string *extend* = " ")

PolygoGroup ツリー、三角形ポリゴン情報の保存。 グループツリーの情報を設定ファイルへ出力。三角形ポリゴン情報をSTL ファイルへ出力。

Parameters

out	*p_config_name*	保存した設定ファイル名の返却用。
in	*stl_format*	TriMeshIO クラスで定義されているSTL ファイルの フォーマット。
in	*extend*	ファイル名に付加する文字列。省略可。省略した 場合は、付加文字列として本メソッド呼び出し時 の年月日時分秒 (YYYYMMDD24hhmmss) を用いる。

Returns

POLYLIB_STAT で定義される値が返る。

Attention

ファイル名命名規約は次の通り。 定義ファイル：polylib_config_ランク番号_付加文字.xml。 STL ファイル：ポリゴングループ名_ランク番号_付加文字.拡張子。

6.11.3.21 template<typename T > char * PolylibNS::Polylib< T >::save_config_file (std::string *rank_no*, std::string *extend*, std::string *format*) [protected]

設定ファイルの保存。 メモリに展開しているグループツリー情報から設定ファイルを生成する。

Parameters

in	*rank_no*	ランク番号
in	*extend*	ファイル名に付加する文字列
in	*format*	TriMeshIO クラスで定義されているSTL ファイルのフォー マット。

Returns

作成した設定ファイルの名称。エラー時はNULL が返る。

6.11.3.22 template<typename T > POLYLIB_STAT PolylibNS::Polylib< T >::save_with_rankno (std::string * *p_config_name,* int *myrank,* int *maxrank,* std::string *extend,* std::string *stl_format,* ID_FORMAT *id_format*) [protected]

PolygoGroup ツリー、三角形ポリゴン情報の保存。 グループツリー情報を設定ファイルへ出力。三角形ポリゴン情報をSTL ファイル へ出力。ID 情報をID ファイルへ出力。ファイル名にランク番号を付加する。

Parameters

out	*p_config_name*	保存した設定ファイル名の返却用。
in	*myrank*	自ランク番号。
in	*maxrank*	最大ランク番号。
in	*extend*	ファイ名に付加される文字列。
in	*stl_format*	STL ファイルフォーマット指定。
in	*id_format*	三角形ID ファイルの出力形式。

Returns

POLYLIB_STAT で定義される値が返る。

Attention

ファイル名命名規約は次の通り。 定義ファイル：polylib_config_ランク番号_付加文字.xml。 STL ファイル：ポリゴングループ名_ランク番号_付加文字.拡張子。 ID ファイル：ポリゴングループ名_ランク番号_付加文字.ID。
MPIPolylib クラスがMPI 環境で利用することを想定している。

6.11.3.23 template<typename T > const Triangle< T > * PolylibNS::Polylib< T >::search_nearest_polygon (std::string *group_name,* const Vec3< T > & *pos*) const

指定した点に最も近い三角形ポリゴンの検索。

Parameters

in	*group_name*	抽出グループ名。
in	*pos*	指定した点。

Returns

検索されたポリゴン

6.11.3.24 template<typename T > std::vector< Triangle< T > * > * PolylibNS::Polylib< T >::search_polygons (std::string *group_name,* Vec3< T > *min_pos,* Vec3< T > *max_pos,* bool *every*) const

三角形ポリゴンの検索。 位置ベクトル min_pos と max_pos により特定される矩形領域に含まれる、三角形ポリゴンを group_name で指定されたグループの下から探索する。

Parameters

in	*group_name*	抽出グループ名。
in	*min_pos*	抽出する矩形領域の最小値。
in	*max_pos*	抽出する矩形領域の最大値。
in	*every*	true:3 頂点が全て検索領域に含まれるものを抽出。 false:3 頂点の一部でも検索領域と重なるものを抽出。

Returns

抽出した三角形ポリゴンの vector。

Attention

返却した三角形ポリゴンは、削除不可。vector は要削除。

6.11.3.25 template<typename T > void PolylibNS::Polylib< T >::set_factory (PolygonGroupFactory< T > * *factory* = NULL)

PolygonGroup クラスを生成するためのFactory クラスを登録。 本メソッドは、独自のFactory クラスを登録しない限り、呼び出し不要である。 コンストラクタで生成したFactory クラスを破棄し、代わりに引数で指定された Factory クラスを登録する。

Parameters

in	*factory*	Factory クラス。

Attention

PolygonGroup を拡張した場合、拡張後のPolygonGroup のFactory クラスを登録する。

6.11.3.26 template<typename T > POLYLIB_STAT PolylibNS::Polylib< T >::setfilepath (std::map< std::string, std::string > & *stl_fname_map*) [protected]

TextParser 内部データに　save した stl ファイルの　"filepath"を書き込む。

　save したSTL ファイルとPolygonGroup の階層は、save_stl_file に　　map を渡し保持してもらう。その map の内容に基づき、TextParser 内部のデータを　変更する.

Parameters

in	*stl_fname_map*	save したSTL ファイルとその階層の map 型データ

Returns

POLYLIB_STAT で定義される値が返る。

6.11.3.27 template<typename T > void PolylibNS::Polylib< T >::show_group_hierarchy (FILE * *fp* = NULL)

2010.10.20 引数FILE * 追加。

グループ階層構造を標準出力に出力。 2010.10.20 引数FILE * 追加。

Parameters

in	*fp*	出力先ファイル。指定されて行ければ、標準出力へ出力する。

Attention

テスト用の関数であり、通常は利用者が用いるものではない。

6.11.3.28 template<typename T > POLYLIB_STAT PolylibNS::Polylib< T >::show_group_info (std::string *group_name*)

グループの情報と配下の三角形ポリゴン情報を標準出力に出力。 親グループ名、自身の名前、STL ファイル名、登録三角形数、3 頂点ベクト ルの座標、法線ベクトルの座標、面積。

Parameters

in	*group_name*	グループ名。

Returns

POLYLIB_STAT で定義される値が返る。

Attention

テスト用の関数であり、通常は利用者が用いるものではない。

6.11.3.29 template<typename T > void PolylibNS::Polylib< T >::show_group_name (PolygonGroup< T > * *p*, std::string *tab*, FILE * *fp*) [protected]

グループ名の表示。 指定されたグループ以下の階層構造をツリー形式で標準出力に出力する。 2010.10.20 引数FILE * 追加。

Parameters

in	*p*	検索の基点となるPolygonGroup のポインタ
in	*tab*	階層の深さを示すスペース
in	*fp*	出力先ファイル。指定されて行ければ、標準出力へ出力する。

6.11.3.30 template<typename T > unsigned int PolylibNS::Polylib< T >::used_memory_size ( )

Polylib が利用中の概算メモリ量を返す

Returns

利用中のメモリ量 (byte)

6.11.4 Member Data Documentation

6.11.4.1 template<typename T> PolygonGroupFactory<T>* PolylibNS::Polylib< T >::m_factory [protected]

自クラスのインスタンス (singleton)

PolygonGroup のファクトリークラス

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/groups/PolygonGroup.h
- /Users/kawanabe/Desktop/Polylib-3.1.0/include/Polylib.h

6.12 PolylibNS::PolylibMoveParams Class Reference

`#include <Polylib.h>`

Public Attributes

- int m_current_step
  現在の計算ステップ番号
- int m_next_step
  移動後の計算ステップ番号
- double m_delta_t
  １計算ステップあたりの時間変異

6.12.1 Detailed Description

クラス:PolylibMoveParams Polylib::move()の引数として利用するパラメタセットクラスです。 本クラスメンバ変数ではパラメタが不足する場合は、継承クラスをユーザ定義 してください。

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/Polylib.h

6.13 PolylibNS::PolylibStat2 Class Reference

`#include <PolylibStat.h>`

Static Public Member Functions

- static std::string String (POLYLIB_STAT stat)

6.13.1 Detailed Description

PolylibStat 文字列出力用クラス

6.13.2 Member Function Documentation

6.13.2.1 static std::string PolylibNS::PolylibStat2::String (POLYLIB_STAT *stat*) [inline], [static]

PolylibStat 文字列出力。

Parameters

in	*stat*	PolylibStat 値。

Returns

PolylibStat 値を文字列化したもの。

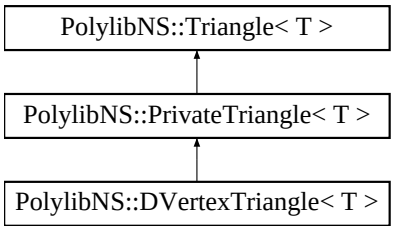
The documentation for this class was generated from the following file:

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/common/PolylibStat.h

6.14 PolylibNS::PrivateTriangle< T > Class Template Reference

`#include <Triangle.h>`

Inheritance diagram for PolylibNS::PrivateTriangle< T >:



Public Member Functions

- PrivateTriangle (Vertex< T > *vertex_ptr[3], int id)
- PrivateTriangle (Vertex< T > *vertex_ptr[3], Vec3< T > normal, int id)
- PrivateTriangle (Vertex< T > *vertex_ptr[3], Vec3< T > normal, T area, int id)
- PrivateTriangle (Triangle< T > tri, int id)
- PrivateTriangle (const PrivateTriangle< T > &tri)
- PrivateTriangle (T *dim, int id)
- virtual void set_id (int id)
- virtual int get_id () const

Protected Attributes

- int m_id

Additional Inherited Members

6.14.1 Detailed Description

template<typename T>class PolylibNS::PrivateTriangle< T >

クラス:PrivateTriangle クラス Polylib 内のデータ保存用の基本クラスです。

6.14.2 Constructor & Destructor Documentation

6.14.2.1 template<typename T> PolylibNS::PrivateTriangle< T >::PrivateTriangle (Vertex< T > * *vertex_ptr[3]*, int *id*) [inline]

コンストラクタ。

Parameters

in	*vertex_ptr*	ポリゴンの頂点へのポインタ。
in	*id*	三角形ポリゴンID。

6.14.2.2 template<typename T> PolylibNS::PrivateTriangle< T >::PrivateTriangle (Vertex< T > * *vertex_ptr[3]*, Vec3< T > *normal*, int *id*) [inline]

コンストラクタ。

Parameters

in	*vertex_ptr*	ポリゴンの頂点へのポインタ。
in	*normal*	法線。
in	*id*	三角形ポリゴンID。

6.14.2.3 template<typename T> PolylibNS::PrivateTriangle< T >::PrivateTriangle (Vertex< T > * *vertex_ptr[3],* Vec3< T > *normal,* T *area,* int *id*) [inline]

コンストラクタ。

Parameters

in	*vertex_ptr*	ポリゴンの頂点へのポインタ。
in	*normal*	法線。
in	*area*	ポリゴンの面積。
in	*id*	三角形ポリゴンID。

6.14.2.4 template<typename T> PolylibNS::PrivateTriangle< T >::PrivateTriangle (Triangle< T > *tri,* int *id*) [inline]

コンストラクタ。

Parameters

in	*tri*	ポリゴン。
in	*id*	三角形ポリゴンID。

6.14.2.5 template<typename T> PolylibNS::PrivateTriangle< T >::PrivateTriangle (const PrivateTriangle< T > & *tri*) [inline]

コンストラクタ。

Parameters

in	*tri*	ポリゴン。

6.14.2.6 template<typename T> PolylibNS::PrivateTriangle< T >::PrivateTriangle (T * *dim,* int *id*) [inline]

コンストラクタ。

Parameters

in	*dim*	ポリゴン頂点座標配列。
in	*id*	三角形ポリゴンID。

6.14.3 Member Function Documentation

6.14.3.1 template<typename T> virtual int PolylibNS::PrivateTriangle< T >::get_id ( ) const [inline], [virtual]

三角形ポリゴンID を返す。

Returns

三角形ポリゴンID。

6.14.3.2 template<typename T> virtual void PolylibNS::PrivateTriangle< T >::set_id (int *id*) [inline], [virtual]

三角形ポリゴンID を設定。

Parameters

in	*id*	三角形ポリゴンID。

6.14.4 Member Data Documentation

6.14.4.1 template<typename T> int PolylibNS::PrivateTriangle< T >::m_id [protected]

PolygonGroup 内で一意となる三角形ポリゴンID。

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/Polygons.h
- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/Triangle.h

6.15 PolylibNS::PrivTriaEqual< T > Struct Template Reference

Public Member Functions

- bool **operator()** (const PrivateTriangle< T > *l, const PrivateTriangle< T > *r) const

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/TriMesh.h

6.16 PolylibNS::PrivTriaLess< T > Struct Template Reference

Public Member Functions

- bool **operator()** (const PrivateTriangle< T > *l, const PrivateTriangle< T > *r) const

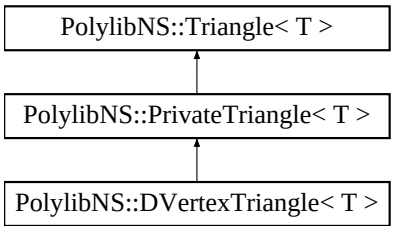
The documentation for this struct was generated from the following file:

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/TriMesh.h

6.17 PolylibNS::Triangle< T > Class Template Reference

`#include <Triangle.h>`

Inheritance diagram for PolylibNS::Triangle< T >:



Public Member Functions

- Triangle ()
- Triangle (Vertex< T > *vertex_ptr[3])
- Triangle (Vertex< T > *vertex_ptr[3], Vec3< T > normal)
- Triangle (Vertex< T > *vertex_ptr[3], Vec3< T > normal, T area)
- virtual void set_vertexes (Vertex< T > *vertex_ptr[3], bool calc_normal, bool calc_area)
- virtual Vertex< T > ** get_vertex () const
- virtual Vec3< T > get_normal () const
- virtual T get_area () const
- virtual void set_exid (int id)
- virtual int get_exid () const
- virtual void set_shell (int val)
- virtual int get_shell () const

Protected Member Functions

- virtual void calc_normal ()
- virtual void calc_area ()

Protected Attributes

- Vertex< T > * m_vertex_ptr [3]
  三角形の頂点座標（反時計回りで並んでいる）。
- Vec3< T > m_normal
  三角形の法線ベクトル。
- T m_area
  三角形の面積。
- int m_exid
  三角形のユーザ定義*ID*
- int m_shell
  三角形のユーザ定義状態変数

6.17.1 Detailed Description

template<typename T>class PolylibNS::Triangle< T >

クラス:Triangle 入出力用インターフェースクラスであり、本ヘッダに対応する.cxx ファイルは存在 しない。

6.17.2 Constructor & Destructor Documentation

6.17.2.1 template<typename T> PolylibNS::Triangle< T >::Triangle ( ) [inline]

コンストラクタ。

6.17.2.2 template<typename T> PolylibNS::Triangle< T >::Triangle (Vertex< T > * *vertex_ptr[3]*) [inline]

コンストラクタ。

Parameters

in	*vertex_ptr*	ポリゴンの頂点。

Attention

面積と法線は vertex を元に自動計算される。

6.17.2.3 template<typename T> PolylibNS::Triangle< T >::Triangle (Vertex< T > * *vertex_ptr[3],* Vec3< T > *normal*) [inline]

コンストラクタ。

Parameters

in	*vertex_ptr*	ポリゴンの頂点。
in	*normal*	法線。

Attention

面積は vertex を元に自動計算される。

6.17.2.4 template<typename T> PolylibNS::Triangle< T >::Triangle (Vertex< T > * *vertex_ptr[3],* Vec3< T > *normal,* T *area*) [inline]

コンストラクタ。

Parameters

in	*vertex_ptr*	ポリゴンの頂点。
in	*normal*	法線。
in	*area*	ポリゴンの面積。

6.17.3 Member Function Documentation

6.17.3.1 template<typename T> virtual void PolylibNS::Triangle< T >::calc_area ( ) [inline], [protected], [virtual]

面積算出。

6.17.3.2 template<typename T> virtual void PolylibNS::Triangle< T >::calc_normal ( ) [inline], [protected], [virtual]

法線ベクトル算出。

6.17.3.3 template<typename T> virtual T PolylibNS::Triangle< T >::get_area ( ) const [inline], [virtual]

面積を取得。

Returns

面積。

6.17.3.4 template<typename T> virtual int PolylibNS::Triangle< T >::get_exid ( ) const [inline], [virtual]

ユーザ定義ID を取得。

Returns

ユーザ定義ID。

6.17.3.5 template<typename T> virtual Vec3<T> PolylibNS::Triangle< T >::get_normal ( ) const [inline], [virtual]

法線ベクトルを取得。

Returns

法線ベクトル。

6.17.3.6 template<typename T> virtual int PolylibNS::Triangle< T >::get_shell ( ) const [inline], [virtual]

ユーザ定義状態変数を取得。

Returns

ユーザ定義状態変数。

6.17.3.7 template<typename T> virtual Vertex<T>** PolylibNS::Triangle< T >::get_vertex ( ) const [inline], [virtual]

vertex の配列を取得。

Returns

vertex の配列。

6.17.3.8 template<typename T> virtual void PolylibNS::Triangle< T >::set_exid (int *id*) [inline], [virtual]

ユーザ定義ID を設定。

6.17.3.9 template<typename T> virtual void PolylibNS::Triangle< T >::set_shell (int *val*) [inline], [virtual]

ユーザ定義状態変数を設定。

6.17.3.10 template<typename T> virtual void PolylibNS::Triangle< T >::set_vertexes (Vertex< T > * *vertex_ptr[3],* bool *calc_normal,* bool *calc_area*) [inline], [virtual]

頂点を設定。

Parameters

in	*vertex_ptr*	三角形の 3 頂点。
in	*calc_normal*	法線ベクトルを再計算するか？
in	*calc_area*	面積を再計算するか？

6.17.4 Member Data Documentation

6.17.4.1 template<typename T> Vertex<T>* PolylibNS::Triangle< T >::m_vertex_ptr[3] [protected]

三角形の頂点座標（反時計回りで並んでいる）。

changed with Vertex and VertexList class since Polylib version 3.0 三角形の頂点座標（反時計回りで並んでいる）。

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/Polygons.h
- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/Triangle.h

6.18 TriangleStruct Struct Reference

`#include <CPolylib.h>`

Public Attributes

- PL_REAL m_vertex [9]
  ３頂点座標
- PL_REAL m_normal [3]
  法線ベクトル
- PL_REAL m_area
  面積

6.18.1 Detailed Description

三角形ポリゴン情報構造体

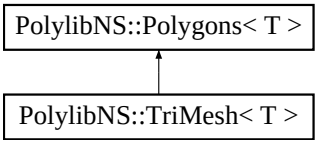
The documentation for this struct was generated from the following file:

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/c_lang/CPolylib.h

6.19 PolylibNS::TriMesh< T > Class Template Reference

`#include <TriMesh.h>`

Inheritance diagram for PolylibNS::TriMesh< T >:



Public Member Functions

- TriMesh ()
- TriMesh (T tolerance)
- ~TriMesh ()
- void init (const std::vector< PrivateTriangle< T > * > *trias)
- void init (const T *vertlist, const int *idlist, const int n_start_tri, const int n_start_id, const unsigned int n_tri)
- virtual void init_dvertex (const T *vertlist, const int *idlist, const T *scalarlist, const T *vectorlist, const int n_start_tri, const int n_start_id, const int n_start_scalar, const int n_start_vector, const unsigned int n_tri, const int n_scalar, const int n_vector)
- void add (const std::vector< PrivateTriangle< T > * > *trias)
- void add (const T *vertlist, const int *idlist, const int n_start_tri, const int n_start_id, const unsigned int n_tri)
- virtual void add_dvertex (const T *vertlist, const int *idlist, const T *scalarlist, const T *vectorlist, const int n_start_tri, const int n_start_id, const int n_start_scalar, const int n_start_vector, const unsigned int n_tri, const int n_scalar, const int n_vector)
- POLYLIB_STAT import (const std::map< std::string, std::string > fmap, T scale=1.0)
- POLYLIB_STAT build ()
- int triangles_num ()
- const std::vector
  < PrivateTriangle< T > * > * search (BBox< T > *bbox, bool every) const
- POLYLIB_STAT search (BBox< T > *bbox, bool every, std::vector< PrivateTriangle< T > * > *tri_list) const
- const std::vector
  < PrivateTriangle< T > * > * linear_search (BBox< T > *q_bbox, bool every) const
- POLYLIB_STAT linear_search (BBox< T > *q_bbox, bool every, std::vector< PrivateTriangle< T > * > *tri_list) const
- const PrivateTriangle< T > * search_nearest (const Vec3< T > &pos) const
- POLYLIB_STAT set_all_exid (const int id) const
- virtual POLYLIB_STAT replace_DVertex (int nscalar, int nvector)
- virtual POLYLIB_STAT prepare_DVertex (int nscalar, int nvector)
- virtual DVertexTriangle< T > * add_DVertex_Triangle (Vec3< T > *v)
- virtual void finalize_DVertex ()
- BBox< T > get_bbox () const
- VertKDT< T > * get_vertkdt () const
- VTree< T > * get_vtree () const
- DVertexManager * DVM () const
- bool hasDVertex () const
- virtual void **print_memory_size** () const

Additional Inherited Members

6.19.1 Detailed Description

template<typename T>class PolylibNS::TriMesh< T >

クラス:TriMesh 三角形ポリゴン集合を管理するクラス（KD 木用に特化したクラス)。

6.19.2 Constructor & Destructor Documentation

6.19.2.1 template<typename T > PolylibNS::TriMesh< T >::TriMesh ( )

コンストラクタ。

6.19.2.2 template<typename T > PolylibNS::TriMesh< T >::TriMesh (T *tolerance*)

コンストラクタ。

Parameters

in	*tolerance*	頂点同一性チェックの基準値

6.19.2.3 template<typename T > PolylibNS::TriMesh< T >::~TriMesh ( )

デストラクタ。

6.19.3 Member Function Documentation

6.19.3.1 template<typename T > void PolylibNS::TriMesh< T >::add (const std::vector< PrivateTriangle< T > * > * *trias*) [virtual]

三角形ポリゴンリストに引数で与えられる三角形の複製を追加する。

Parameters

in	*trias*	設定する三角形ポリゴンリスト。

Attention

m_id が重複するインスタンスは追加されない。
KD 木の再構築は行わない。

Implements PolylibNS::Polygons< T >.

6.19.3.2 template<typename T > void PolylibNS::TriMesh< T >::add (const T * *vertlist*, const int * *idlist*, const int *n_start_tri*, const int *n_start_id*, const unsigned int *n_tri*) [virtual]

三角形ポリゴンリストに引数で与えられる三角形の複製を追加する。

Parameters

in	*vertlist*	設定する三角形ポリゴン頂点リスト。
in	*idlist*	三角形の id。
in	*n_start_tri*	vertlist の頂点開始位置
in	*n_start_id*	idlist の id 開始位置
in	*n_tri*	加える三角形の数

Implements PolylibNS::Polygons< T >.

6.19.3.3 template<typename T > void PolylibNS::TriMesh< T >::add_dvertex (const T * *vertlist*, const int * *idlist*, const T * *scalarlist*, const T * *vectorlist*, const int *n_start_tri*, const int *n_start_id*, const int *n_start_scalar*, const int *n_start_vector*, const unsigned int *n_tri*, const int *n_scalar*, const int *n_vector*) [virtual]

三角形ポリゴンリストに引数で与えられる三角形 (DVertexTriangle) を追加する。

Parameters

in	*vertlist*	設定する三角形ポリゴン頂点リスト。
in	*idlist*	三角形の id。
in	*scalarlist*	設定するスカラーデータのリスト

in	*vectorlist*	設定するベクターデータのリスト
in	*n_start_tri*	vertlist の頂点開始位置
in	*n_start_id*	idlist の id 開始位置
in	*n_start_scalar*	scalarlist の開始位置
in	*n_start_vector*	vectorlist の開始位置
in	*n_tri*	加える三角形の数
in	*n_scalar*	頂点あたりのスカラーデータの数
in	*n_vector*	頂点あたりのベクターデータの数

Implements PolylibNS::Polygons< T >.

6.19.3.4 template<typename T > DVertexTriangle< T > * PolylibNS::TriMesh< T >::add_DVertex_Triangle (Vec3< T > * *v*) [virtual]

DVertex 追加作成用

Parameters

in	*v*	頂点座標（3 点）

Returns

polygon への pointer

Reimplemented from PolylibNS::Polygons< T >.

6.19.3.5 template<typename T > POLYLIB_STAT PolylibNS::TriMesh< T >::build ( ) [virtual]

Polygons クラスに含まれる全ポリゴン情報からKD 木を作成する。

Returns

POLYLIB_STAT で定義される値が返る。

TriMesh クラスに含まれる全三角形ポリゴンを外包するBoundingBox を計算

Implements PolylibNS::Polygons< T >.

6.19.3.6 template<typename T> DVertexManager* PolylibNS::TriMesh< T >::DVM ( ) const [inline]

DVertexManager

Returns

KD 木クラス。

6.19.3.7 template<typename T> virtual void PolylibNS::TriMesh< T >::finalize_DVertex ( ) [inline], [virtual]

DVertex 追加作成後の重複頂点削除

Reimplemented from PolylibNS::Polygons< T >.

6.19.3.8 template<typename T> BBox<T> PolylibNS::TriMesh< T >::get_bbox ( ) const [inline], [virtual]

TriMesh クラスが管理しているBoundingBox を返す。

Implements PolylibNS::Polygons< T >.

6.19.3.9 template<typename T> VertKDT<T>* PolylibNS::TriMesh< T >::get_vertkdt ( ) const [inline], [virtual]

KD 木クラスを取得。

Returns

KD 木クラス。

Implements PolylibNS::Polygons< T >.

6.19.3.10 template<typename T> VTree<T>* PolylibNS::TriMesh< T >::get_vtree ( ) const [inline], [virtual]

KD 木クラスを取得。

Returns

KD 木クラス。

Implements PolylibNS::Polygons< T >.

6.19.3.11 template<typename T> bool PolylibNS::TriMesh< T >::hasDVertex ( ) const [inline], [virtual]

hasDVertex

Returns

KD 木クラス。

Implements PolylibNS::Polygons< T >.

6.19.3.12 template<typename T > POLYLIB_STAT PolylibNS::TriMesh< T >::import (const std::map< std::string, std::string > *fmap*, T *scale* = 1.0) [virtual]

ファイルからデータの初期化。

Parameters

in	*fmap*	ファイル名、ファイルフォーマット。

Returns

PLSTAT_OK=成功/false=失敗

Implements PolylibNS::Polygons< T >.

6.19.3.13 template<typename T > void PolylibNS::TriMesh< T >::init (const std::vector< PrivateTriangle< T > * > * *trias*) [virtual]

TriMesh クラスで管理する三角形ポリゴンリストを初期化し、引数で与えら れる三角形ポリゴンリストを設定する。 三角形ポリゴン用のメモリ領域は、Polylib 内で新たに確保される。

Parameters

in	*trias*	設定する三角形ポリゴンリスト。

Implements PolylibNS::Polygons< T >.

6.19.3.14 template<typename T > void PolylibNS::TriMesh< T >::init (const T * *vertlist*, const int * *idlist*, const int *n_start_tri*, const int *n_start_id*, const unsigned int *n_tri*) [virtual]

TriMesh クラスで管理する三角形ポリゴンリストを初期化し、引数で与えら れる三角形ポリゴンリストを設定する。 三角形ポリゴン用のメモリ領域は、TriMesh 内で新たに確保する。

Parameters

in	*vertlist*	設定する三角形ポリゴン頂点リスト。
in	*idlist*	三角形の id。
in	*n_start_tri*	vertlist の頂点開始位置
in	*n_start_id*	idlist の id 開始位置
in	*n_tri*	加える三角形の数

Implements PolylibNS::Polygons< T >.

6.19.3.15 template<typename T > void PolylibNS::TriMesh< T >::init_dvertex (const T * *vertlist*, const int * *idlist*, const T * *scalarlist*, const T * *vectorlist*, const int *n_start_tri*, const int *n_start_id*, const int *n_start_scalar*, const int *n_start_vector*, const unsigned int *n_tri*, const int *n_scalar*, const int *n_vector*) [virtual]

三角形ポリゴンリストを初期化し、引数で与えら れる三角形ポリゴンリストを設定する。 三角形ポリゴン用のメモリ領域は、TriMesh 内で新たに確保する。

Parameters

in	*vertlist*	設定する三角形ポリゴン頂点リスト。
in	*idlist*	三角形の id。
in	*scalarlist*	設定するスカラーデータのリスト
in	*vectorlist*	設定するベクターデータのリスト
in	*n_start_tri*	vertlist の頂点開始位置
in	*n_start_id*	idlist の id 開始位置
in	*n_start_scalar*	scalarlist の開始位置
in	*n_start_vector*	vectorlist の開始位置
in	*n_tri*	加える三角形の数
in	*n_scalar*	頂点あたりのスカラーデータの数
in	*n_vector*	頂点あたりのベクターデータの数

Implements PolylibNS::Polygons< T >.

6.19.3.16 template<typename T > const std::vector< PrivateTriangle< T > * > * PolylibNS::TriMesh< T >::linear_search (BBox< T > * *q_bbox*, bool *every*) const [virtual]

線形探索により、指定矩形領域に含まれる三角形ポリゴンを抽出する。

Parameters

in	*q_bbox*	検索範囲を示す矩形領域。
in	*every*	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。

Returns

抽出したポリゴンリストのポインタ。

Attention

三角形ポリゴンのメモリ領域は新たにPolylib 内で確保される。
MPIPolylib 内での利用が目的なので、ユーザは使用しないこと。

Implements PolylibNS::Polygons< T >.

6.19.3.17 template<typename T > POLYLIB_STAT PolylibNS::TriMesh< T >::linear_search (BBox< T > * *q_bbox*, bool *every*, std::vector< PrivateTriangle< T > * > * *tri_list*) const [virtual]

線形探索により、指定矩形領域に含まれるポリゴンを抽出する。

Parameters

in	*q_bbox*	検索範囲を示す矩形領域。
in	*every*	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。
in,out	*tri_list*	抽出した三角形ポリゴンリストへのポインタ。

Returns

POLYLIB_STAT で定義される値が返る。

Attention

tri_list で戻される三角形ポリゴンのポインタは、Polylib 内で 保持されるアドレス値なので、ユーザは delete しないで下さい。
オーバーロードメソッドあり。

Implements PolylibNS::Polygons< T >.

6.19.3.18 template<typename T > POLYLIB_STAT PolylibNS::TriMesh< T >::prepare_DVertex (int *nscalar*, int *nvector*) [virtual]

Vertex -> DVertex への準備

Parameters

in	*nscalar*	スカラーデータ数
in	*nvector*	ベクトルデータ数

Reimplemented from PolylibNS::Polygons< T >.

6.19.3.19 template<typename T > POLYLIB_STAT PolylibNS::TriMesh< T >::replace_DVertex (int *nscalar*, int *nvector*) [virtual]

Vertex -> DVertex へのリプレース

Parameters

in	*nscalar*	スカラーデータ数
in	*nvector*	ベクトルデータ数

Reimplemented from PolylibNS::Polygons< T >.

6.19.3.20 template<typename T > const std::vector< PrivateTriangle< T > * > * PolylibNS::TriMesh< T >::search (BBox< T > * *bbox*, bool *every*) const [virtual]

KD 木探索により、指定矩形領域に含まれる三角形ポリゴンを抽出する。

Parameters

in	*bbox*	検索範囲を示す矩形領域。
in	*every*	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。

Returns

抽出したポリゴンリストのポインタ。

Attention

三角形ポリゴンのメモリ領域は新たにPolylib 内で確保される。
MPIPolylib 内での利用が目的なので、ユーザは使用しないこと。
オーバーロードメソッドあり。

Implements PolylibNS::Polygons< T >.

6.19.3.21 template<typename T > POLYLIB_STAT PolylibNS::TriMesh< T >::search (BBox< T > * *bbox*, bool *every*, std::vector< PrivateTriangle< T > * > * *tri_list*) const [virtual]

KD 木探索により、指定矩形領域に含まれるポリゴンを抽出する。

Parameters

in	*bbox*	検索範囲を示す矩形領域
in	*every*	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。
in,out	*tri_list*	抽出した三角形ポリゴンリストへのポインタ。

Returns

POLYLIB_STAT で定義される値が返る。

Attention

tri_list で戻される三角形ポリゴンのポインタは、Polylib 内で 保持されるアドレス値なので、ユーザは delete しないで下さい。
オーバーロードメソッドあり。

Implements PolylibNS::Polygons< T >.

6.19.3.22 template<typename T > const PrivateTriangle< T > * PolylibNS::TriMesh< T >::search_nearest (const Vec3< T > & *pos*) const [virtual]

KD 木探索により、指定位置に最も近いポリゴンを検索する。

Parameters

in	*pos*	指定位置

Returns

検索されたポリゴン

Implements PolylibNS::Polygons< T >.

6.19.3.23 template<typename T > POLYLIB_STAT PolylibNS::TriMesh< T >::set_all_exid (const int *id*) const [virtual]

配下の全ポリゴンの m_exid 値を指定値にする。

Parameters

in	*id*	指定値

Implements PolylibNS::Polygons< T >.

6.19.3.24 template<typename T > int PolylibNS::TriMesh< T >::triangles_num ( ) [virtual]

TriMesh クラスが管理している三角形ポリゴン数を返す。

Implements PolylibNS::Polygons< T >.

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/TriMesh.h

6.20 PolylibNS::TriMeshIO Class Reference

`#include <TriMeshIO.h>`

Static Public Member Functions

- template<typename T >
  static POLYLIB_STAT load (VertexList< T > *vertex_list, std::vector< PrivateTriangle< T > * > *tri_list, const std::map< std::string, std::string > &fmap, T scale=1.0)
- template<typename T >
  static POLYLIB_STAT save (VertexList< T > *vertex_list, std::vector< PrivateTriangle< T > * > *tri_list, std::string fname, std::string fmt="")
- static std::string input_file_format (const std::string &filename)

Static Public Attributes

- static const std::string FMT_STL_A
  アスキーファイル
- static const std::string FMT_STL_AA
  アスキーファイル
- static const std::string FMT_STL_B
  バイナリファイル
- static const std::string FMT_STL_BB
  バイナリファイル
- static const std::string FMT_OBJ_A
  ascii
- static const std::string FMT_OBJ_AA
  ascii
- static const std::string FMT_OBJ_B
  binary
- static const std::string FMT_OBJ_BB
  binary
- static const std::string FMT_VTK_A
  vtk ascii
- static const std::string FMT_VTK_B
  vtk binary
- static const std::string DEFAULT_FMT
  TrimeshIO.cxx **で定義している値**

6.20.1 Detailed Description

クラス:TriMeshIO 三角形ポリゴン入出力管理。

6.20.2 Member Function Documentation

6.20.2.1 static std::string PolylibNS::TriMeshIO::input_file_format (const std::string & *filename*) [static]

ファイル名を元に入力ファイルのフォーマットを取得する。

Parameters

in	*filename*	入力ファイル名。

Returns

判定したファイルフォーマット。

Attention

ファイル拡張子が"stl"の場合、ファイルを読み込んで判定する。

6.20.2.2 template<typename T > POLYLIB_STAT PolylibNS::TriMeshIO::load (VertexList< T > * *vertex_list,* std::vector< PrivateTriangle< T > * > * *tri_list,* const std::map< std::string, std::string > & *fmap,* T *scale* = 1.0) [static]

STL /OBJ ファイルを読み込み、tri_list にセットする。

Parameters

in,out	*vertex_list*	頂点リストの領域。
in,out	*tri_list*	三角形ポリゴンリストの領域。
in	*fmap*	ファイル名、ファイルフォーマットのセット。

Returns

POLYLIB_STAT で定義される値が返る。

6.20.2.3 template<typename T > POLYLIB_STAT PolylibNS::TriMeshIO::save (VertexList< T > * *vertex_list,* std::vector< PrivateTriangle< T > * > * *tri_list,* std::string *fname,* std::string *fmt* = " ") [static]

tri_list の内容をSTL 形式でファイルへ保存。

Parameters

in	*tri_list*	三角形ポリゴンのリスト (出力内容)。
in	*fname*	ファイル名。
in	*fmt*	ファイルフォーマット。

Returns

POLYLIB_STAT で定義される値が返る。

6.20.3 Member Data Documentation

6.20.3.1 const std::string PolylibNS::TriMeshIO::FMT_STL_A [static]

アスキーファイル

STL ファイルのフォーマット種別

Attention

STL ファイルの拡張子とは異なるので注意すること。

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/file_io/TriMeshIO.h

6.21 PolylibNS::TryMesh< T > Class Template Reference

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/groups/PolygonGroup.h

6.22 PolylibNS::Vec2< T > Class Template Reference

`#include <Vec2.h>`

Public Member Functions

- **Vec2** (T v=0)
- **Vec2** (T _x, T _y)
- **Vec2** (const T v[2])
- Vec2< T > & **assign** (T _x, T _y)
- **operator T** * ()
- **operator const T** * () const
- T * **ptr** ()
- const T * **ptr** () const
- T & **operator[]** (int i)
- const T & **operator[]** (int i) const
- Vec2< T > & **operator+=** (const Vec2< T > &v)
- Vec2< T > & **operator-=** (const Vec2< T > &v)
- Vec2< T > & **operator*=** (const Vec2< T > &v)
- Vec2< T > & **operator/=** (const Vec2< T > &v)
- Vec2< T > & **operator*=** (T s)
- Vec2< T > & **operator/=** (T s)
- Vec2< T > **operator+** (const Vec2< T > &v) const
- Vec2< T > **operator-** (const Vec2< T > &v) const
- Vec2< T > **operator*** (const Vec2< T > &v) const
- Vec2< T > **operator/** (const Vec2< T > &v) const
- Vec2< T > **operator*** (T s) const
- Vec2< T > **operator/** (T s) const
- Vec2< T > **operator-** () const
- bool **operator==** (const Vec2< T > &v) const
- bool **operator!=** (const Vec2< T > &v) const

- T **lengthSquared** () const
- T **length** () const
- Vec2< T > & **normalize** ()
- Vec2< T > & **normalize** (T *len)
- T **average** () const

Static Public Member Functions

- static Vec2< T > **xaxis** ()
- static Vec2< T > **yaxis** ()

Public Attributes

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

6.22.1 Detailed Description

template<typename T>class PolylibNS::Vec2< T >

クラス:Vec2<T>

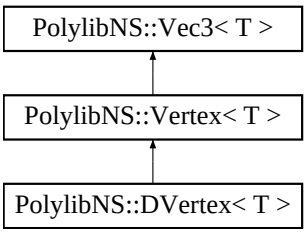
The documentation for this class was generated from the following file:

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/common/Vec2.h

6.23 PolylibNS::Vec3< T > Class Template Reference

`#include <Vec3.h>`

Inheritance diagram for PolylibNS::Vec3< T >:



Public Member Functions

- **Vec3** (T v=0)
- **Vec3** (T _x, T _y, T _z)
- **Vec3** (const T v[3])
- Vec3< T > & **assign** (T _x, T _y, T _z)
- **operator T** * ()
- **operator const T** * () const
- T * **ptr** ()
- const T * **ptr** () const
- T & **operator[]** (const AxisEnum &axis)
- const T & **operator[]** (const AxisEnum &axis) const

- Vec3< T > & **operator+=** (const Vec3< T > &v)
- Vec3< T > & **operator-=** (const Vec3< T > &v)
- Vec3< T > & **operator*=** (const Vec3< T > &v)
- Vec3< T > & **operator/=** (const Vec3< T > &v)
- Vec3< T > & **operator*=** (T s)
- Vec3< T > & **operator/=** (T s)
- Vec3< T > **operator+** (const Vec3< T > &v) const
- Vec3< T > **operator-** (const Vec3< T > &v) const
- Vec3< T > **operator*** (const Vec3< T > &v) const
- Vec3< T > **operator/** (const Vec3< T > &v) const
- Vec3< T > **operator*** (T s) const
- Vec3< T > **operator/** (T s) const
- Vec3< T > **operator-** () const
- bool **operator==** (const Vec3< T > &v) const
- bool **operator!=** (const Vec3< T > &v) const
- T **lengthSquared** () const
- T **length** () const
- Vec3< T > & **normalize** ()
- Vec3< T > & **normalize** (T *len)
- float **average** () const

Static Public Member Functions

- static Vec3< T > **xaxis** ()
- static Vec3< T > **yaxis** ()
- static Vec3< T > **zaxis** ()

Public Attributes

- T **t** [3]

6.23.1 Detailed Description

template<typename T>class PolylibNS::Vec3< T >

クラス:Vec3<T>

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/common/Vec3.h

6.24 PolylibNS::VElement< T > Class Template Reference

`#include <VTree.h>`

Public Member Functions

- VElement (PrivateTriangle< T > *tri)
- PrivateTriangle< T > * get_triangle ()
- Vec3< T > get_pos () const
- BBox< T > get_bbox () const

6.24.1 Detailed Description

template<typename T>class PolylibNS::VElement< T >

クラス:VElement KD 木構造の要素クラスです。

6.24.2 Constructor & Destructor Documentation

6.24.2.1 template<typename T > PolylibNS::VElement< T >::VElement (PrivateTriangle< T > * *tri*)

コンストラクタ。

Parameters

in	*tri*	ポリゴン情報のポインタ。

Attention

ポインタを格納するが、参照のみ。delete は行わない。

6.24.3 Member Function Documentation

6.24.3.1 template<typename T> BBox<T> PolylibNS::VElement< T >::get_bbox ( ) const [inline]

Bounding box of this triangle

6.24.3.2 template<typename T> Vec3<T> PolylibNS::VElement< T >::get_pos ( ) const [inline]

Center position of bbox on triangle.

6.24.3.3 template<typename T> PrivateTriangle<T>* PolylibNS::VElement< T >::get_triangle ( ) [inline]

triangle。

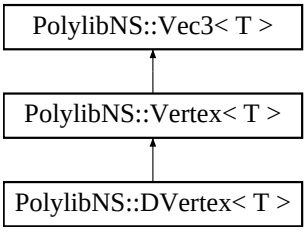
The documentation for this class was generated from the following file:

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/VTree.h

6.25 PolylibNS::Vertex< T > Class Template Reference

`#include <Vertex.h>`

Inheritance diagram for PolylibNS::Vertex< T >:



Public Member Functions

- Vertex ()
- Vertex (const Vec3< T > &vec)
- Vertex (T x, T y, T z)
- virtual T & operator[] (const AxisEnum &axis)
- virtual const T & operator[] (const AxisEnum &axis) const
- virtual T distanceSquared (Vertex v)
- virtual T distance (Vertex v)

  距離

Additional Inherited Members

6.25.1 Detailed Description

template<typename T>class PolylibNS::Vertex< T >

クラス:vertex polygon の頂点クラス。

6.25.2 Constructor & Destructor Documentation

6.25.2.1 template<typename T> PolylibNS::Vertex< T >::Vertex ( ) [inline]

コンストラクタ

6.25.2.2 template<typename T> PolylibNS::Vertex< T >::Vertex (const Vec3< T > & *vec*) [inline]

コンストラクタ

Parameters

in	*vec*	頂点ベクトル

6.25.2.3 template<typename T> PolylibNS::Vertex< T >::Vertex (T *x*, T *y*, T *z*) [inline]

コンストラクタ

Parameters

in	*x*	座標
in	*y*	座標
in	*z*	座標

6.25.3 Member Function Documentation

6.25.3.1 template<typename T> virtual T PolylibNS::Vertex< T >::distanceSquared (Vertex< T > *v*) [inline], [virtual]

2 乗距離

6.25.3.2 template<typename T> virtual T& PolylibNS::Vertex< T >::operator[] (const AxisEnum & *axis*) [inline], [virtual]

index アクセス

6.25.3.3 template<typename T> virtual const T& PolylibNS::Vertex< T >::operator[] (const AxisEnum & *axis*) const [inline], [virtual]

index アクセス

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/Vertex.h

6.26 PolylibNS::VertexList< T > Class Template Reference

`#include <VertexList.h>`

Public Member Functions

- **VertexList** (T tolerance)
- VertexList (VertKDT< T > *vkdt, T tolerance)
- ~VertexList ()
- void setKDT (VertKDT< T > *vkdt)
- VertKDT< T > * getKDT ()
  Vertex 用*KD* 木
- const std::vector< Vertex< T > * > * get_vertex_lists ()
  Vertex の格納場所へのポインタ
- void vtx_add_nocheck (Vertex< T > *v)
  Vertex の追加 同一性チェック無し。
- int vtx_add_i (Vertex< T > *v)
  Vertex の追加、*m_vertex_list* の *index* を返す。同一性チェック済み。
- Vertex< T > * vtx_add (Vertex< T > *v)
  Vertex の追加、その頂点のポインタを示す。同一性チェック済み。
- Vertex< T > * vtx_add_KDT (Vertex< T > *v)
  Vertex の追加、その頂点のポインタを示す。同一性チェック済み。
- Vertex< T > * ith (int i)
  i 番目*Vertex* を取り出す。
- Vertex< T > * ith (long i)
  i 番目*Vertex* を取り出す。
- long ith (Vertex< T > *vtx_ptr) const
  i 番目*Vertex* を取り出す。
- std::vector< Vertex< T > * >::size_type size ()
- void set_bbox ()
  bbox を*VertKDT* へ設定
- BBox< T > get_bbox () const
  bbox を*VertKDT* を取得
- void prepare_num_out ()
- std::vector< Vertex< T > * >::size_type vtx_index (Vertex< T > *v)

- void index_map_clear ()
  三角形ポリゴンの面出力時、頂点の番号を取得する場合の、番号のテーブルを削除する。
- void set_tolerance (const T tol)
  基準値を設定する
- T tolerance () const
  基準
- void print () const
  プリント
- VertexList ()
  コンストラクタ　基準値
- T get_tolerance ()
  コンストラクタ　基準値
- POLYLIB_STAT vertex_compaction (std::map< Vertex< T > *, Vertex< T > * > *vertex_map)
  重複頂点の削除
- void vtx_clear ()
  Vertex **の解放**

6.26.1 Detailed Description

template<typename T>class PolylibNS::VertexList< T >

クラス:vertex_list polygon の頂点クラスVertex を収めるクラス。

6.26.2 Constructor & Destructor Documentation

6.26.2.1 template<typename T> PolylibNS::VertexList< T >::VertexList (VertKDT< T > * *vkdt*, T *tolerance*) [inline]

コンストラクタ

6.26.2.2 template<typename T> PolylibNS::VertexList< T >::~VertexList ( ) [inline]

デストラクタ
Vertex は削除。削除する場合は、vtx_clear を呼ぶ。 VertKDT は削除しない。削除する場合は、 外部から

6.26.3 Member Function Documentation

6.26.3.1 template<typename T> T PolylibNS::VertexList< T >::get_tolerance ( ) [inline]

コンストラクタ　基準値
基準値の取得

6.26.3.2 template<typename T> void PolylibNS::VertexList< T >::prepare_num_out ( ) [inline]

三角形ポリゴンの面出力時、頂点の番号を取得する場合に、番号のテーブルを準備する。
後始末は index_map_clear()で行う。

6.26.3.3 template<typename T> void PolylibNS::VertexList< T >::setKDT (VertKDT< T > * *vkdt*) [inline]

Vertex 用KD 木のセット

Parameters

in	*vkdt*	利用するKD 木のポインタ

6.26.3.4 template<typename T> std::vector<Vertex<T>*>::size_type PolylibNS::VertexList< T >::size ( ) [inline]

頂点の数

Returns

頂点の数

6.26.3.5 template<typename T> std::vector<Vertex<T>*>::size_type PolylibNS::VertexList< T >::vtx_index (Vertex< T > * *v*) [inline]

三角形ポリゴンの面出力時、頂点の番号を取得する

Parameters

in	*Vertex<T>*	のポインタ

Returns

頂点番号

Attention

prepare_num_out() を呼び出す。

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/TriMesh.h
- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/VertexList.h

6.27 PolylibNS::VertKDT< T > Class Template Reference

`#include <VertKDT.h>`

Public Member Functions

- VertKDT (int max_elem, const BBox< T > bbox, std::vector< Vertex< T > * > *vert_list)
- VertKDT (int max_elem)
- ~VertKDT ()
- void destroy ()
- const Vertex< T > * search_nearest (const Vec3< T > &pos) const
- const Vertex< T > * search_nearest_recursive (VertKDTNode< T > *vn, const Vec3< T > &pos) const
- unsigned int memory_size ()
- BBox< T > get_root_bbox ()

  root node の*BBox* を返す
- void set_root_bbox (const BBox< T > &box)

  root node の*BBox* を 設定する。
- POLYLIB_STAT add (Vertex< T > *v)

- POLYLIB_STAT add2 (Vertex< T > *v)
- POLYLIB_STAT make_upper (Vertex< T > *v)
- POLYLIB_STAT create (const BBox< T > bbox, std::vector< Vertex< T > * > *vert_list)
- int **n_create** ()

6.27.1 Detailed Description

template<typename T>class PolylibNS::VertKDT< T >

クラス:VertKDT リーフをVertex とするKD 木のクラスです。

6.27.2 Constructor & Destructor Documentation

6.27.2.1 template<typename T > PolylibNS::VertKDT< T >::VertKDT (int *max_elem*, const BBox< T > *bbox*, std::vector< Vertex< T > * > * *vert_list*)

コンストラクタ。

Parameters

in	*max_elem*	最大要素数。
in	*bbox*	VertKDT の box 範囲。
in	*vert_list*	木構造の元になる vertex のリスト

6.27.2.2 template<typename T> PolylibNS::VertKDT< T >::VertKDT (int *max_elem*) [inline]

コンストラクタ。

Parameters

in	*max_elem*	最大要素数。

6.27.2.3 template<typename T > PolylibNS::VertKDT< T >::~VertKDT ( )

デストラクタ。

6.27.3 Member Function Documentation

6.27.3.1 template<typename T > POLYLIB_STAT PolylibNS::VertKDT< T >::add (Vertex< T > * *v*)

kd 木に vertex を追加する。

Parameters

in	*v*	頂点クラス

Returns

ステータス

6.27.3.2 template<typename T > POLYLIB_STAT PolylibNS::VertKDT< T >::add2 (Vertex< T > * *v*)

kd 木に vertex を追加する。 BBox に v が入っていなければ、make_upper を用いて再帰的に拡大する。 最後は add を用いる。

Parameters

in	*v*	頂点クラス

Returns

ステータス

6.27.3.3 template<typename T> POLYLIB_STAT PolylibNS::VertKDT< T >::create (const BBox< T > *bbox*, std::vector< Vertex< T > * > * *vert_list*) [inline]

KD 木　再構築用関数

Parameters

in	*bbox*	VertKDT の box 範囲。
in	*vertex_list*	木構造の元になるVertex のリスト。

Returns

POLYLIB_STAT で定義される値が返る。

6.27.3.4 template<typename T > void PolylibNS::VertKDT< T >::destroy ( )

木構造を消去する。

6.27.3.5 template<typename T > POLYLIB_STAT PolylibNS::VertKDT< T >::make_upper (Vertex< T > * *v*)

BBox を拡大する。v が入ると思われる方向に拡大する。 ある軸について、v がBBox の min より小さい時、に左方向に拡大。 それ以外は右方向へ拡大する。

Parameters

in	*v*	頂点クラス

Returns

ステータス

6.27.3.6 template<typename T > unsigned int PolylibNS::VertKDT< T >::memory_size ( )

KD 木クラスが利用しているメモリ量を返す。

Returns

利用中のメモリ量 (byte)

6.27.3.7 template<typename T > const Vertex< T > * PolylibNS::VertKDT< T >::search_nearest (const Vec3< T > & *pos*) const

KD 木探索により、指定位置に最も近いVertex を検索する。

Parameters

in	*pos*	指定位置

Returns

検索されたVertex

6.27.3.8 template<typename T > const Vertex< T > * PolylibNS::VertKDT< T >::search_nearest_recursive (VertKDTNode< T > * *vn*, const Vec3< T > & *pos*) const

KD 木探索により、指定位置に最も近いVertex を検索する。

Parameters

in	*vn*	検索対象のノードへのポインタ。
in	*pos*	指定位置

Returns

検索されたVertex

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/TriMesh.h
- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/VertKDT.h

6.28 PolylibNS::VertKDTElem< T > Class Template Reference

`#include <VertKDT.h>`

Public Member Functions

- VertKDTElem (Vertex< T > *vertex)
- Vertex< T > * get_vertex ()
- Vec3< T > * get_pos () const

6.28.1 Detailed Description

template<typename T>class PolylibNS::VertKDTElem< T >

クラス:VertKDTElem VertexKD 木の要素クラスです。

6.28.2 Constructor & Destructor Documentation

6.28.2.1 template<typename T> PolylibNS::VertKDTElem< T >::VertKDTElem (Vertex< T > * *vertex*) [inline]

コンストラクタ。

Parameters

in	*@attention*	ポインタを格納するが、参照のみ。delete は行わない。

6.28.3 Member Function Documentation

6.28.3.1 template<typename T> Vec3<T>* PolylibNS::VertKDTElem< T >::get_pos ( ) const [inline]

return vertex.

6.28.3.2 template<typename T> Vertex<T>* PolylibNS::VertKDTElem< T >::get_vertex ( ) [inline]

return vertex.

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/VertKDT.h

6.29 PolylibNS::VertKDTNode< T > Class Template Reference

`#include <VertKDT.h>`

Public Member Functions

- VertKDTNode ()
- ~VertKDTNode ()
- void split (const int &max_elem)
- bool is_leaf () const
- BBox< T > get_bbox () const
- void set_bbox (const BBox< T > &bbox)
- BBox< T > get_bbox_search () const
- void set_bbox_search (const VertKDTElem< T > *p)
- VertKDTNode< T > * get_left ()
- VertKDTNode< T > * get_right ()
- AxisEnum get_axis () const
- void set_axis (const AxisEnum axis)
- std::vector< VertKDTElem< T > * > & get_vlist ()
- void set_element (VertKDTElem< T > *elm)
- int get_elements_num () const
- void set_left_node (VertKDTNode< T > *lnode)
- void set_right_node (VertKDTNode< T > *rnode)

6.29.1 Detailed Description

template<typename T>class PolylibNS::VertKDTNode< T >

クラス:VertKDTNode VertexKD 木の要素クラスです。

6.29.2 Constructor & Destructor Documentation

6.29.2.1 template<typename T > PolylibNS::VertKDTNode< T >::VertKDTNode ( )

コンストラクタ。

6.29.2.2 template<typename T > PolylibNS::VertKDTNode< T >::~VertKDTNode ( )

デストラクタ。

6.29.3 Member Function Documentation

6.29.3.1 template<typename T> AxisEnum PolylibNS::VertKDTNode< T >::get_axis ( ) const [inline]

Axis を取得。

Returns

axis。

6.29.3.2 template<typename T> BBox<T> PolylibNS::VertKDTNode< T >::get_bbox ( ) const [inline]

BBox の値を取得

Returns

bbox。

6.29.3.3 template<typename T> BBox<T> PolylibNS::VertKDTNode< T >::get_bbox_search ( ) const [inline]

検索用BBox<T> を取得。

Returns

検索用 bbox。

6.29.3.4 template<typename T> int PolylibNS::VertKDTNode< T >::get_elements_num ( ) const [inline]

ノードが所持する要素の数を取得。

Returns

要素数。

6.29.3.5 template<typename T> VertKDTNode<T>* PolylibNS::VertKDTNode< T >::get_left ( ) [inline]

左のNode を取得。

Returns

左のNode。

6.29.3.6 template<typename T> VertKDTNode<T>* PolylibNS::VertKDTNode< T >::get_right ( ) [inline]

右のNode を取得。

Returns

右のNode。

6.29.3.7 template<typename T> std::vector<VertKDTElem<T>*>& PolylibNS::VertKDTNode< T >::get_vlist ( ) [inline]

要素のリストを取得。

Returns

要素のリスト。

6.29.3.8 template<typename T> bool PolylibNS::VertKDTNode< T >::is_leaf ( ) const [inline]

ノードがリーフかどうか判定する。

Returns

true=リーフ/false=リーフでない。

6.29.3.9 template<typename T> void PolylibNS::VertKDTNode< T >::set_axis (const AxisEnum *axis*) [inline]

Axis を設定。

Parameters

in	*axis*。	

6.29.3.10 template<typename T> void PolylibNS::VertKDTNode< T >::set_bbox (const BBox< T > & *bbox*) [inline]

BBox の値を設定

Parameters

in	*bbox*。	

6.29.3.11 template<typename T> void PolylibNS::VertKDTNode< T >::set_bbox_search (const VertKDTElem< T > * *p*) [inline]

このノードのBounding Box を引数で与えられる要素を含めた大きさに変更する。

Parameters

in	*p*	要素。

6.29.3.12 template<typename T> void PolylibNS::VertKDTNode< T >::set_element (VertKDTElem< T > * *elm*) [inline]

木の要素を設定。

Parameters

in	*elm*。	

6.29.3.13 template<typename T> void PolylibNS::VertKDTNode< T >::set_left_node (VertKDTNode< T > * *lnode*) [inline]

下位の left ノードを設定。

6.29.3.14 template<typename T> void PolylibNS::VertKDTNode< T >::set_right_node (VertKDTNode< T > * *rnode*) [inline]

下位の right ノードを設定。

6.29.3.15 template<typename T > void PolylibNS::VertKDTNode< T >::split (const int & *max_elem*)

ノードを２つの子供ノードに分割する。

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/VertKDT.h

6.30 PolylibNS::VNode< T > Class Template Reference

`#include <VTree.h>`

Public Member Functions

- VNode ()
- ~VNode ()
- void split (const int &max_elem)
- bool is_leaf () const
- BBox< T > get_bbox () const
- void set_bbox (const BBox< T > &bbox)
- BBox< T > get_bbox_search () const
- void set_bbox_search (const VElement< T > *p)
- VNode< T > * get_left ()
- VNode< T > * get_right ()
- AxisEnum get_axis () const
- void set_axis (const AxisEnum axis)
- std::vector< VElement< T > * > & get_vlist ()
- void set_element (VElement< T > *elm)
- int get_elements_num () const

6.30.1 Detailed Description

template<typename T>class PolylibNS::VNode< T >

VNode クラス KD 木構造のノードクラスです。

6.30.2 Constructor & Destructor Documentation

6.30.2.1 template<typename T > PolylibNS::VNode< T >::VNode ( )

コンストラクタ。

6.30.2.2 template<typename T > PolylibNS::VNode< T >::~VNode ( )

デストラクタ。

6.30.3 Member Function Documentation

6.30.3.1 template<typename T> AxisEnum PolylibNS::VNode< T >::get_axis ( ) const [inline]

Axis を取得。

Returns

axis。

6.30.3.2 template<typename T> BBox<T> PolylibNS::VNode< T >::get_bbox ( ) const [inline]

BBox<T> の値を取得。

Returns

bbox。

6.30.3.3 template<typename T> BBox<T> PolylibNS::VNode< T >::get_bbox_search ( ) const [inline]

検索用BBox<T> を取得。

Returns

検索用 bbox。

6.30.3.4 template<typename T> int PolylibNS::VNode< T >::get_elements_num ( ) const [inline]

ノードが所持する要素の数を取得。

Returns

要素数。

6.30.3.5 template<typename T> VNode<T>* PolylibNS::VNode< T >::get_left ( ) [inline]

左のNode を取得。

Returns

左のNode。

6.30.3.6 template<typename T> VNode<T>* PolylibNS::VNode< T >::get_right ( ) [inline]

右のNode を取得。

Returns

右のNode。

6.30.3.7 template<typename T> std::vector<VElement<T>*>& PolylibNS::VNode< T >::get_vlist ( ) [inline]

要素のリストを取得。

Returns

要素のリスト。

6.30.3.8 template<typename T> bool PolylibNS::VNode< T >::is_leaf ( ) const [inline]

ノードがリーフかどうかの判定結果。

Returns

true=リーフ/false=リーフでない。

6.30.3.9 template<typename T> void PolylibNS::VNode< T >::set_axis (const AxisEnum *axis*) [inline]

Axis を設定。

Parameters

in	*axis*。	

6.30.3.10 template<typename T> void PolylibNS::VNode< T >::set_bbox (const BBox< T > & *bbox*) [inline]

BBox<T> の値を設定。

Parameters

in	*bbox*。	

6.30.3.11 template<typename T> void PolylibNS::VNode< T >::set_bbox_search (const VElement< T > * *p*) [inline]

このノードのBounding Box を引数で与えられる要素を含めた大きさに変更する。

Parameters

in	*p*	要素。

6.30.3.12 template<typename T> void PolylibNS::VNode< T >::set_element (VElement< T > * *elm*) [inline]

木の要素を設定。

Parameters

in	*elm*。	

6.30.3.13 template<typename T > void PolylibNS::VNode< T >::split (const int & *max_elem*)

ノードを２つの子供ノードに分割する。

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/VTree.h

6.31 PolylibNS::VTree< T > Class Template Reference

`#include <VTree.h>`

Public Member Functions

- VTree (int max_elem, const BBox< T > bbox, std::vector< PrivateTriangle< T > * > *tri_list)
- ~VTree ()
- void destroy ()
- std::vector< PrivateTriangle < T > * > * search (BBox< T > *bbox, bool every) const
- POLYLIB_STAT search (BBox< T > *bbox, bool every, std::vector< PrivateTriangle< T > * > *tri_list) const
- const PrivateTriangle< T > * search_nearest (const Vec3< T > &pos) const
- const PrivateTriangle< T > * search_nearest_recursive (VNode< T > *vn, const Vec3< T > &pos) const
- unsigned int memory_size ()

6.31.1 Detailed Description

template<typename T>class PolylibNS::VTree< T >

クラス:VTree リーフを三角形ポリゴンとするKD 木クラスです。

6.31.2 Constructor & Destructor Documentation

6.31.2.1 template<typename T > PolylibNS::VTree< T >::VTree (int *max_elem*, const BBox< T > *bbox*, std::vector< PrivateTriangle< T > * > * *tri_list*)

コンストラクタ。

Parameters

in	*max_elem*	最大要素数。
in	*bbox*	VTree の box 範囲。
in	*tri_list*	木構造の元になるポリゴンのリスト。

6.31.2.2 template<typename T > PolylibNS::VTree< T >::~VTree ( )

デストラクタ。

6.31.3 Member Function Documentation

6.31.3.1 template<typename T > void PolylibNS::VTree< T >::destroy ( )

木構造を消去する。

6.31.3.2 template<typename T > unsigned int PolylibNS::VTree< T >::memory_size ( )

KD 木クラスが利用しているメモリ量を返す。

Returns

利用中のメモリ量 (byte)

6.31.3.3 template<typename T > std::vector< PrivateTriangle< T > * > * PolylibNS::VTree< T >::search (BBox< T > * *bbox*, bool *every*) const

KD 木探索により、指定矩形領域に含まれる三角形ポリゴンを抽出する。

Parameters

in	*bbox*	検索範囲を示す矩形領域。
in	*every*	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。

Returns

抽出したポリゴンリストのポインタ。

Attention

MPIPolylib 用のメソッドなので、ユーザは利用しないで下さい。
オーバーロードメソッドあり。

#undef DEBUG_VTREE

6.31.3.4 template<typename T > POLYLIB_STAT PolylibNS::VTree< T >::search (BBox< T > * *bbox*, bool *every*, std::vector< PrivateTriangle< T > * > * *tri_list*) const

KD 木探索により、指定矩形領域に含まれるポリゴンを抽出する。

Parameters

in	*bbox*	検索範囲を示す矩形領域。
in	*every*	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。
in,out	*tri_list*	抽出した三角形ポリゴンリストへのポインタ。

Returns

POLYLIB_STAT で定義される値が返る。

Attention

オーバーロードメソッドあり。

6.31.3.5 template<typename T > const PrivateTriangle< T > * PolylibNS::VTree< T >::search_nearest (const Vec3< T > & *pos*) const

KD 木探索により、指定位置に最も近いポリゴンを検索する。

Parameters

in	*pos*	指定位置

Returns

検索されたポリゴン

6.31.3.6 template<typename T > const PrivateTriangle< T > * PolylibNS::VTree< T >::search_nearest_recursive (VNode< T > * *vn*, const Vec3< T > & *pos*) const

KD 木探索により、指定位置に最も近いポリゴンを検索する。

Parameters

in	*vn*	検索対象のノードへのポインタ。
in	*pos*	指定位置

Returns

検索されたポリゴン

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

- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/TriMesh.h
- /Users/kawanabe/Desktop/Polylib-3.1.0/include/polygons/VTree.h

Chapter 7

File Documentation

7.1 /Users/kawanabe/Desktop/Polylib-3.1.0/include/Version.h File Reference

Macros

- #define PL_VERSION_NO "3.1.0"
- #define PL_REVISION "20131030_1200"

7.1.1 Detailed Description

Polylib バージョン情報のヘッダーファイル

7.1.2 Macro Definition Documentation

7.1.2.1 #define PL_REVISION "20131030_1200"

POLYLIB ライブラリのリビジョン

7.1.2.2 #define PL_VERSION_NO "3.1.0"

POLYLIB ライブラリのバージョン