ゲームグラフィックス特論

構築: Doxygen 1.8.6

2017 年 05 月 20 日 (土) 21 時 41 分 44 秒

Contents

Chapter 1

名前空間索引

1.1 名前空間一覧

全名前空間の一覧です。

Chapter 2

階層索引

2.1 クラス階層

クラス階層一覧です。大雑把に文字符号順で並べられています。

Chapter 3

クラス索引

3.1 クラス一覧

クラス・構造体・共用体・インターフェースの一覧です。

Chapter 4

ファイル索引

4.1 ファイル一覧

ファイル一覧です。

Chapter 5

名前空間詳解

5.1 gg 名前空間

クラス

- class GgMatrix
  変換行列.
- class GgQuaternion
  四元数.
- class GgTrackball
  簡易トラックボール処理.
- class GgCounter
  参照カウンタ.
- class GgAttribute
  属性データ.
- class GgTexture
  テクスチャ.
- class GgNormalTexture
  法線マップ.
- class GgBuffer
  バッファオブジェクト.
- class GgShape
  形状データの基底クラス.
- class GgPoints
  ポイント.
- class GgTriangles
  三角形で表した形状データ *(Arrays* **形式***)*.
- class GgElements
  三角形で表した形状データ *(Elements* **形式***)*.
- class GgShader
  シェーダの基底クラス.
- class GgPointShader
  ポイントのシェーダ.
- class GgSimpleShader
  三角形に単純な陰影付けを行うシェーダ.
- class GgObj
  Wavefront OBJ **形式のファイル**.

関数

- void ggInit ()

  ゲームグラフィックス特論の都合にもとづく初期化を行う.
- void ggError (const char *name=NULL, unsigned int line=0)

  OpenGL **のエラーをチェックする**.
- void ggFBOError (const char *name=NULL, unsigned int line=0)

  FBO **のエラーをチェックする**.
- bool ggSaveTga (GLsizei sx, GLsizei sy, unsigned int depth, const void *buffer, const char *name)

  配列の内容を *TGA* **ファイルに保存する**.
- bool ggSaveColor (const char *name)

  カラーバッファの内容を *TGA* **ファイルに保存する**.
- bool ggSaveDepth (const char *name)

  デプスバッファの内容を *TGA* **ファイルに保存する**.
- GLubyte * ggLoadTga (const char *name, GLsizei *width, GLsizei *height, GLenum *format)

  TGA **ファイル** *(8/16/24/32bit)* **をメモリに読み込む**.
- GLuint ggLoadTexture (GLsizei width, GLsizei height, GLenum internal, GLenum format=GL_RGBA, const GLvoid *image=NULL)

  テクスチャメモリを確保して画像データをテクスチャとして読み込む.
- GLuint ggLoadImage (const char *name, GLenum internal=0)

  TGA **画像ファイルをテクスチャとして読み込む**.
- GLuint ggLoadHeight (const char *name, float nz, GLenum internal=GL_RGBA)

  高さマップ用の *TGA* **画像ファイルの読み込んで法線マップを作成する**.
- bool ggLoadObj (const char *name, GLuint &nv, GLfloat(*&pos)[3], GLfloat(*&norm)[3], GLuint &nf, GLuint(*&face)[3], bool normalize=false)

  三角形分割された *OBJ* **ファイルを読み込む** *(Elements* **形式***)*.
- bool ggLoadObj (const char *name, GLuint &ng, GLuint(*&group)[2], GLfloat(*&amb)[4], GLfloat(*&diff)[4], GLfloat(*&spec)[4], GLfloat *&shi, GLuint &nv, GLfloat(*&pos)[3], GLfloat(*&norm)[3], bool normalize=false)

  三角形分割された *OBJ* **ファイルと** *MTL* **ファイルを読み込む** *(Arrays* **形式***)*
- GLuint ggCreateShader (const char *vsrc, const char *fsrc=NULL, const char *gsrc=NULL, GLint nvarying=0, const char *const varyings[]=NULL, const char *vtext="vertex shader", const char *ftext="fragment shader", const char *gtext="geometry shader")

  シェーダのソースプログラムの文字列を読み込んでプログラムオブジェクトを作成する.
- GLuint ggLoadShader (const char *vert, const char *frag=NULL, const char *geom=NULL, GLint nvarying=0, const char *const varyings[]=NULL)

  シェーダのソースファイルを読み込んでプログラムオブジェクトを作成する.
- GLfloat ggLength3 (const GLfloat *a)

  3 **要素の長さ**.
- GLfloat ggLength4 (const GLfloat *a)

  4 **要素の長さ**.
- GLfloat ggDot3 (const GLfloat *a, const GLfloat *b)

  3 **要素の内積**.
- void ggCross (GLfloat *c, const GLfloat *a, const GLfloat *b)

  3 **要素の外積**.
- GLfloat ggDot4 (const GLfloat *a, const GLfloat *b)

  4 **要素の内積**
- GgMatrix ggIdentity ()

  単位行列を返す.
- GgMatrix ggTranslate (GLfloat x, GLfloat y, GLfloat z, GLfloat w=1.0f)

  平行移動の変換行列を返す.
- GgMatrix ggTranslate (const GLfloat *t)

  平行移動の変換行列を返す.

- GgMatrix ggScale (GLfloat x, GLfloat y, GLfloat z, GLfloat w=1.0f)
  拡大縮小の変換行列を返す.
- GgMatrix ggScale (const GLfloat *s)
  拡大縮小の変換行列を返す.
- GgMatrix ggRotateX (GLfloat a)
  x **軸中心の回転の変換行列を返す**.
- GgMatrix ggRotateY (GLfloat a)
  y **軸中心の回転の変換行列を返す**.
- GgMatrix ggRotateZ (GLfloat a)
  z **軸中心の回転の変換行列を返す**.
- GgMatrix ggRotate (GLfloat x, GLfloat y, GLfloat z, GLfloat a)
  (x, y, z) **方向のベクトルを軸とする回転の変換行列を乗じた結果を返す**.
- GgMatrix ggRotate (const GLfloat *r, GLfloat a)
  r **方向のベクトルを軸とする回転の変換行列を乗じた結果を返す**.
- GgMatrix ggRotate (const GLfloat *r)
  r **方向のベクトルを軸とする回転の変換行列を乗じた結果を返す**.
- GgMatrix ggLookat (GLfloat ex, GLfloat ey, GLfloat ez, GLfloat tx, GLfloat ty, GLfloat tz, GLfloat ux, GLfloat uy, GLfloat uz)
  ビュー変換行列を返す.
- GgMatrix ggLookat (const GLfloat *e, const GLfloat *t, const GLfloat *u)
  ビュー変換行列を返す.
- GgMatrix ggOrthogonal (GLfloat left, GLfloat right, GLfloat bottom, GLfloat top, GLfloat zNear, GLfloat zFar)
  直交投影変換行列を返す.
- GgMatrix ggFrustum (GLfloat left, GLfloat right, GLfloat bottom, GLfloat top, GLfloat zNear, GLfloat zFar)
  透視透視投影変換行列を返す.
- GgMatrix ggPerspective (GLfloat fovy, GLfloat aspect, GLfloat zNear, GLfloat zFar)
  画角を指定して透視投影変換行列を返す.
- GgMatrix ggTranspose (const GgMatrix &m)
  転置行列を返す.
- GgMatrix ggInvert (const GgMatrix &m)
  逆行列を返す.
- GgMatrix ggNormal (const GgMatrix &m)
  法線変換行列を返す.
- GgQuaternion ggQuaternion (GLfloat x, GLfloat y, GLfloat z, GLfloat w)
  四元数を返す
- GgQuaternion ggQuaternion (const GLfloat *a)
  四元数を返す
- GgQuaternion ggIdentityQuaternion ()
  単位四元数を返す
- GgQuaternion ggMatrixQuaternion (const GLfloat *a)
  回転の変換行列 *m* **を表す四元数を返す**.
- GgQuaternion ggMatrixQuaternion (const GgMatrix &m)
  回転の変換行列 *m* **を表す四元数を返す**.
- GgMatrix ggQuaternionMatrix (const GgQuaternion &q)
  四元数 *q* **の回転の変換行列を返す**.
- GgMatrix ggQuaternionTransposeMatrix (const GgQuaternion &q)
  四元数 *q* **の回転の転置した変換行列を返す**.
- GgQuaternion ggRotateQuaternion (GLfloat x, GLfloat y, GLfloat z, GLfloat a)
  (x, y, z) **を軸として角度** *a* **回転する四元数を返す**.
- GgQuaternion ggRotateQuaternion (const GLfloat *v, GLfloat a)
  (v[0], v[1], v[2]) **を軸として角度** *a* **回転する四元数を返す**.

- GgQuaternion ggRotateQuaternion (const GLfloat *v)
  (v[0], v[1], v[2]) を軸として角度 *v[3]* 回転する四元数を返す.
- GgQuaternion ggEulerQuaternion (GLfloat heading, GLfloat pitch, GLfloat roll)
  オイラー角 *(heading, pitch, roll)* で与えられた回転を表す四元数を返す.
- GgQuaternion ggEulerQuaternion (const GLfloat *e)
  オイラー角 *(e[0], e[1], e[2])* で与えられた回転を表す四元数を返す.
- GgQuaternion ggSlerp (const GLfloat *a, const GLfloat *b, GLfloat t)
  二つの四元数の球面線形補間の結果を返す.
- GgQuaternion ggSlerp (const GgQuaternion &q, const GgQuaternion &r, GLfloat t)
  二つの四元数の球面線形補間の結果を返す.
- GgQuaternion ggSlerp (const GgQuaternion &q, const GLfloat *a, GLfloat t)
  二つの四元数の球面線形補間の結果を返す.
- GgQuaternion ggSlerp (const GLfloat *a, const GgQuaternion &q, GLfloat t)
  二つの四元数の球面線形補間の結果を返す.
- GLfloat ggNorm (const GgQuaternion &q)
  四元数のノルムを返す.
- GgQuaternion ggNormalize (const GgQuaternion &q)
  正規化した四元数を返す.
- GgQuaternion ggConjugate (const GgQuaternion &q)
  共役四元数を返す.
- GgQuaternion ggInvert (const GgQuaternion &q)
  四元数の逆元を求める.
- GgPoints * ggPointsCube (GLuint nv, GLfloat length=1.0f, GLfloat cx=0.0f, GLfloat cy=0.0f, GLfloat cz=0.0f)
  点群を立方体状に生成する.
- GgPoints * ggPointsSphere (GLuint nv, GLfloat radius=0.5f, GLfloat cx=0.0f, GLfloat cy=0.0f, GLfloat cz=0.0f)
  点群を球状に生成する.
- GgTriangles * ggRectangle (GLfloat width=1.0f, GLfloat height=1.0f)
  矩形状に *2* 枚の三角形を生成する.
- GgTriangles * ggEllipse (GLfloat width=1.0f, GLfloat height=1.0f, GLuint slices=16)
  楕円状に三角形を生成する.
- GgTriangles * ggArraysObj (const char *name, bool normalize=false)
  Wavefront OBJ ファイルを読み込む *(Arrays* 形式*)*
- GgElements * ggElementsObj (const char *name, bool normalize=false)
  Wavefront OBJ ファイル を読み込む *(Elements* 形式*)*.
- GgElements * ggElementsMesh (int slices, int stacks, const GLfloat(*pos)[3], const GLfloat(*norm)[3]=NULL)
  メッシュ形状を作成する *(Elements* 形式*)*.
- GgElements * ggElementsSphere (GLfloat radius=1.0f, int slices=16, int stacks=8)
  球状に三角形データを生成する *(Elements* 形式*)*.

5.1.1 関数詳解

5.1.1.1 gg::GgTriangles * gg::ggArraysObj (const char * *name*, bool *normalize* = `false`)

Wavefront OBJ ファイルを読み込む (Arrays 形式)

Wavefront OBJ ファイルを読み込む (Arrays 形式).

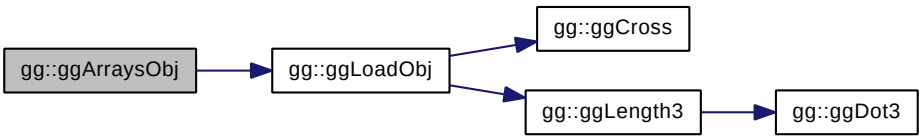
三角形分割された Wavefront OBJ ファイルを読み込んで GgArrays 形式の三角形データを生成する.

引数

name	ファイル名.
normalize	true なら大きさを正規化.

gg.cpp の 7703 行目に定義があります。

呼び出し関係図:



5.1.1.2 GgQuaternion gg::ggConjugate (const GgQuaternion & *q*) [inline]

共役四元数を返す.

引数

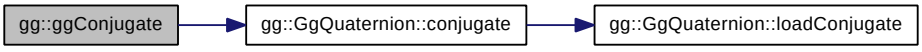
q	GgQuaternion 型の四元数.

戻り値

四元数 q の共役四元数.

gg.h の 4599 行目に定義があります。

呼び出し関係図:



5.1.1.3 GLuint gg::ggCreateShader (const char * *vsrc*, const char * *fsrc* = NULL, const char * *gsrc* = NULL, GLint *nvarying* = 0, const char *const *varyings[]* = NULL, const char * *vtext* = "vertex shader", const char * *ftext* = "fragment shader", const char * *gtext* = "geometry shader")

シェーダのソースプログラムの文字列を読み込んでプログラムオブジェクトを作成する.

引数

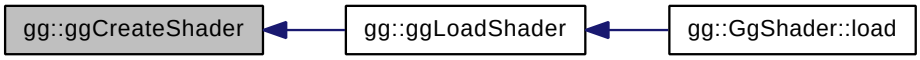
vsrc	バーテックスシェーダのソースプログラムの文字列.
fsrc	フラグメントシェーダのソースプログラムの文字列 (NULL なら不使用).
gsrc	ジオメトリシェーダのソースプログラムの文字列 (NULL なら不使用).
nvarying	フィードバックする varying 変数の数 (0 なら不使用).
varyings	フィードバックする varying 変数のリスト (NULL なら不使用).
vtext	バーテックスシェーダのコンパイル時のメッセージに追加する文字列.
ftext	フラグメントシェーダのコンパイル時のメッセージに追加する文字列.
gtext	ジオメトリシェーダのコンパイル時のメッセージに追加する文字列.

戻り値

シェーダプログラムのプログラム名 (作成できなければ 0).

gg.cpp の 6598 行目に定義があります。

被呼び出し関係図:



5.1.1.4 void gg::ggCross (GLfloat * c, const GLfloat * a, const GLfloat * b) [inline]

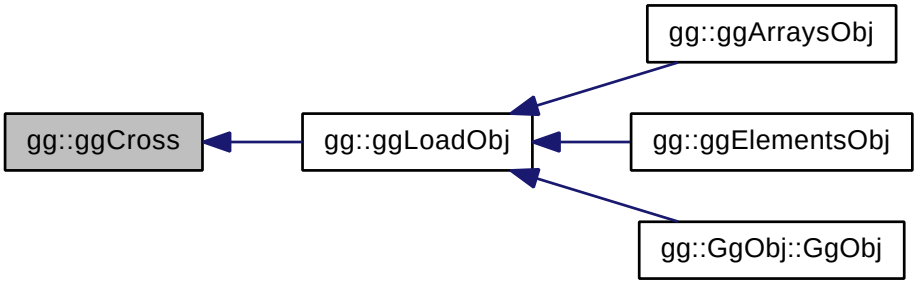
3 要素の外積.

引数

a	GLfloat 型の 3 要素の配列.
b	GLfloat 型の 3 要素の配列.
c	結果を格納する GLfloat 型の 3 要素の配列.

gg.h の 2784 行目に定義があります。

被呼び出し関係図:



5.1.1.5 GLfloat gg::ggDot3 (const GLfloat * a, const GLfloat * b) [inline]

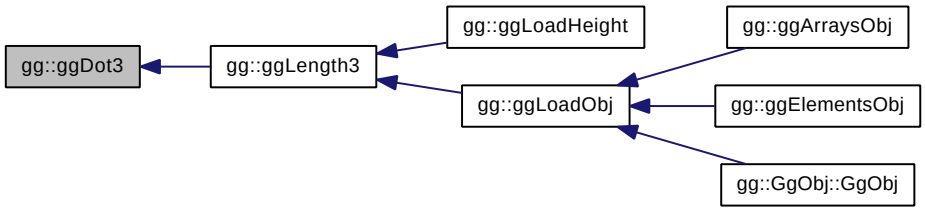
3 要素の内積.

引数

a	GLfloat 型の 3 要素の配列.
b	GLfloat 型の 3 要素の配列.

gg.h の 2772 行目に定義があります。

被呼び出し関係図:



5.1.1.6 GLfloat gg::ggDot4 (const GLfloat * *a*, const GLfloat * *b*) [inline]

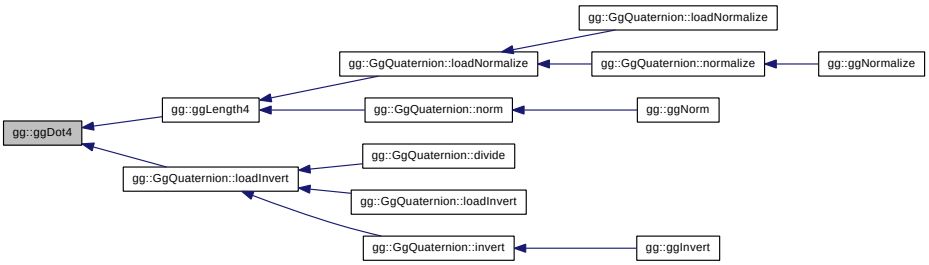
4 要素の内積

引数

a	GLfloat 型の 4 要素の配列.
b	GLfloat 型の 4 要素の配列.

gg.h の 2797 行目に定義があります。

被呼び出し関係図:



5.1.1.7 gg::GgElements * gg::ggElementsMesh (int *slices*, int *stacks*, const GLfloat(*) *pos[3]*, const GLfloat(*) *norm[3]* = NULL)

メッシュ形状を作成する (Elements 形式).

メッシュ状に GgElements 形式の三角形データを生成する.

引数

slices	メッシュの横方向の分割数.
stacks	メッシュの縦方向の分割数.
pos	メッシュの頂点の位置.
norm	メッシュの頂点の法線ベクトル.

gg.cpp の 7769 行目に定義があります。

被呼び出し関係図:

5.1.1.8 gg::GgElements * gg::ggElementsObj (const char * *name*, bool *normalize* = false)

Wavefront OBJ ファイル を読み込む (Elements 形式).

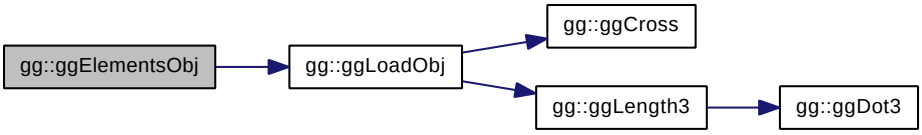
三角形分割された Wavefront OBJ ファイル を読み込んで GgElements 形式の三角形データを生成する.

引数

name	ファイル名.
normalize	true なら大きさを正規化.

gg.cpp の 7738 行目に定義があります。

呼び出し関係図:



5.1.1.9 gg::GgElements * gg::ggElementsSphere (GLfloat *radius* = 1.0f, int *slices* = 16, int *stacks* = 8)

球状に三角形データを生成する (Elements 形式).

球状に三角形データを生成する (Elements 形式).

球状に GgElements 形式の三角形データを生成する.

引数

radius	球の半径.
slices	球の経度方向の分割数.
stacks	球の緯度方向の分割数.

球状に GgElements 形式の三角形データを生成する.

引数

radius	球の半径.
slices	球の経度方向の分割数.
stacks	球の緯度方向の分割数.

gg.cpp の 7870 行目に定義があります。

呼び出し関係図:

5.1.1.10 gg::GgTriangles * gg::ggEllipse (GLfloat *width* = 1.0f, GLfloat *height* = 1.0f, GLuint *slices* = 16)

楕円状に三角形を生成する.

引数

width	楕円の幅.
height	楕円の高さ.
slices	楕円の分割数.

gg.cpp の 7670 行目に定義があります。

5.1.1.11 void gg::ggError (const char * *name* = NULL, unsigned int *line* = 0)

OpenGL のエラーをチェックする.

OpenGL の API を呼び出し直後に実行すればエラーのあるときにメッセージを表示する.

引数

msg	エラー発生時に標準エラー出力に出力する文字列. NULL なら何も出力しない.

gg.cpp の 5169 行目に定義があります。

5.1.1.12 GgQuaternion gg::ggEulerQuaternion (GLfloat *heading*, GLfloat *pitch*, GLfloat *roll*) [inline]

オイラー角 (heading, pitch, roll) で与えられた回転を表す四元数を返す.

引数

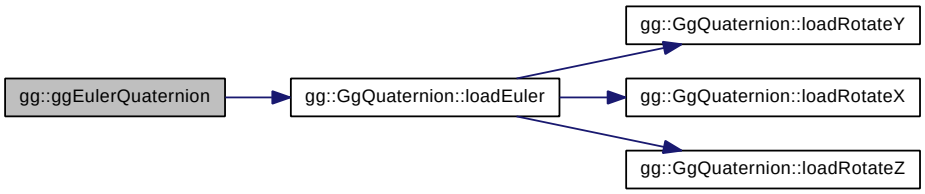
heading	y 軸中心の回転角.
pitch	x 軸中心の回転角.
roll	z 軸中心の回転角.

戻り値

回転を表す四元数.

gg.h の 4525 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:

5.1.1.13 GgQuaternion gg::ggEulerQuaternion (const GLfloat * *e*) [inline]

オイラー角 (e[0], e[1], e[2]) で与えられた回転を表す四元数を返す.

引数

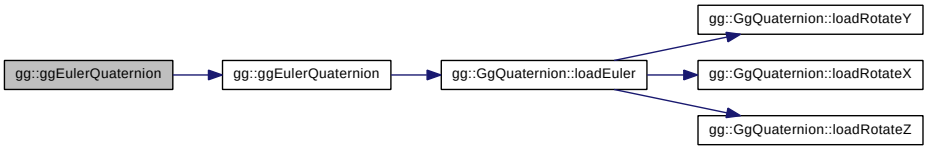
e	オイラー角を表す GLfloat 型の 3 要素の配列 (heading, pitch, roll).

戻り値

回転を表す四元数.

gg.h の 4534 行目に定義があります。

呼び出し関係図:



5.1.1.14 void gg::ggFBOError (const char * *name* = NULL, unsigned int *line* = 0)

FBO のエラーをチェックする.

FBO の API を呼び出し直後に実行すればエラーのあるときにメッセージを表示する.

引数

msg	エラー発生時に標準エラー出力に出力する文字列. NULL なら何も出力しない.

gg.cpp の 5213 行目に定義があります。

5.1.1.15 GgMatrix gg::ggFrustum (GLfloat *left*, GLfloat *right*, GLfloat *bottom*, GLfloat *top*, GLfloat *zNear*, GLfloat *zFar*) [inline]

透視透視投影変換行列を返す.

引数

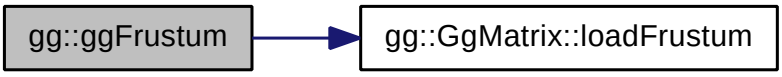
left	ウィンドウの左端の位置.
right	ウィンドウの右端の位置.
bottom	ウィンドウの下端の位置.
top	ウィンドウの上端の位置.
zNear	視点から前方面までの位置.
zFar	視点から後方面までの位置.

戻り値

求めた透視投影変換行列.

gg.h の 3626 行目に定義があります。

呼び出し関係図:



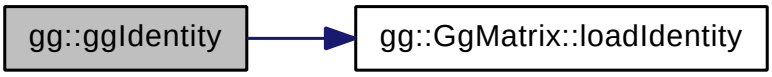
5.1.1.16 GgMatrix gg::ggIdentity ( ) [inline]

単位行列を返す.

戻り値

単位行列

gg.h の 3460 行目に定義があります。
呼び出し関係図:



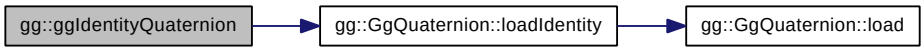
5.1.1.17 GgQuaternion gg::ggIdentityQuaternion ( ) [inline]

単位四元数を返す

戻り値

単位四元数.

gg.h の 4446 行目に定義があります。
呼び出し関係図:



5.1.1.18 void gg::ggInit ( )

ゲームグラフィックス特論の都合にもとづく初期化を行う.
Windows で OpenGL 1.2 以降の API を有効化する.
gg.cpp の 2604 行目に定義があります。

5.1.1.19 GgMatrix gg::ggInvert (const GgMatrix & *m*) [inline]

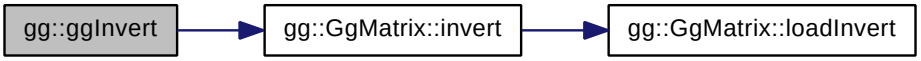
逆行列を返す.
引数

m	元の変換行列.

戻り値

m の逆行列.

gg.h の 3658 行目に定義があります。
呼び出し関係図:



5.1.1.20 GgQuaternion gg::ggInvert (const GgQuaternion & *q*) [inline]

四元数の逆元を求める.

引数

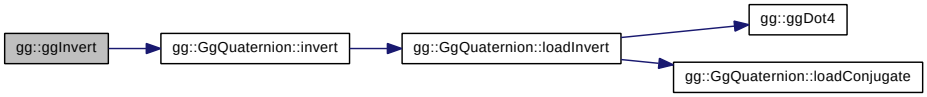
q	GgQuaternion 型の四元数.

戻り値

四元数 q の逆元.

gg.h の 4607 行目に定義があります。

呼び出し関係図:



5.1.1.21 GLfloat gg::ggLength3 (const GLfloat * *a*)

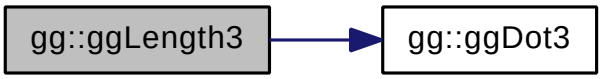
3 要素の長さ.

引数

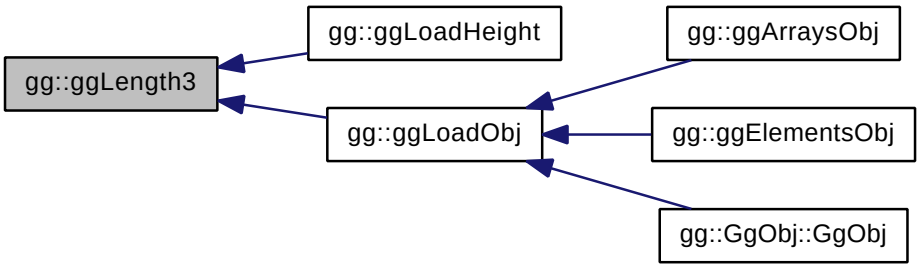
a	GLfloat 型の 3 要素の配列.

gg.cpp の 6753 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:



5.1.1.22 GLfloat gg::ggLength4 (const GLfloat * *a*)

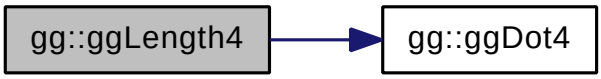
4 要素の長さ.

引数

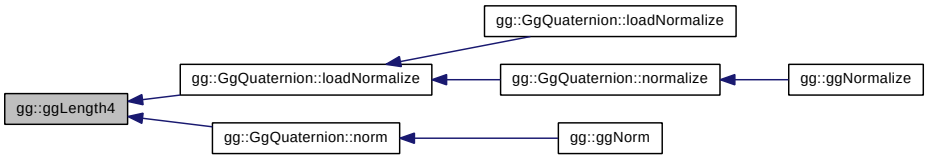
a	GLfloat 型の 4 要素の配列.

gg.cpp の 6763 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:



5.1.1.23 GLuint gg::ggLoadHeight (const char * *name*, float *nz*, GLenum *internal* = GL_RGBA)

高さマップ用の TGA 画像ファイルの読み込んで法線マップを作成する.

TGA 画像ファイルの高さマップ読み込んでテクスチャメモリに法線マップを作成する.

引数

name	読み込むファイル名.
nz	法線ベクトルの z 成分の割合.
internal	glTexImage2D() に指定するテクスチャの内部フォーマット.

戻り値

読み込みに成功すれば true, 失敗すれば false.

引数

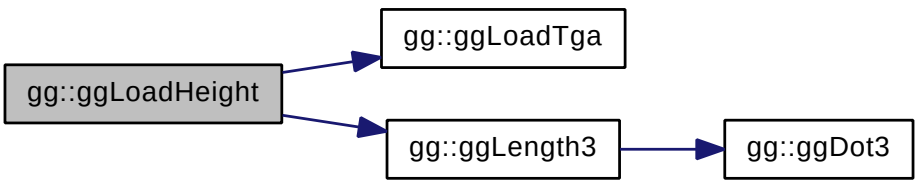
name	TGA ファイル名.
nz	作成した法線ベクトルの z 成分の割合.
internal	テクスチャの内部フォーマット.

戻り値

テクスチャオブジェクト名.

gg.cpp の 5614 行目に定義があります。

呼び出し関係図:



5.1.1.24 GLuint gg::ggLoadImage (const char * *name*, GLenum *internal* = 0)

TGA 画像ファイルをテクスチャとして読み込む.

TGA ファイルをテクスチャメモリに読み込む.

引数

name	読み込むファイル名.
internal	glTexImage2D() に指定するテクスチャの内部フォーマット. 0 なら外部フォーマットに合わせる.

戻り値

読み込みに成功すれば true, 失敗すれば false.

引数

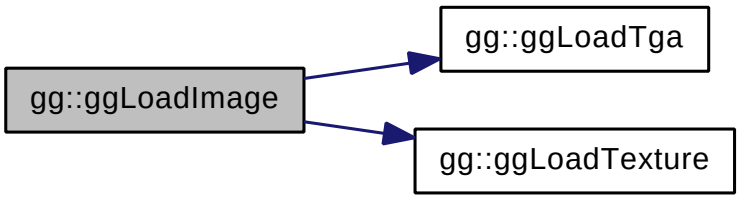
name	TGA ファイル名.
internal	テクスチャの内部フォーマット.

戻り値

テクスチャオブジェクト名.

gg.cpp の 5565 行目に定義があります。

呼び出し関係図:



5.1.1.25 bool gg::ggLoadObj (const char * *name*, GLuint & *nv*, GLfloat(*&) *pos[3]*, GLfloat(*&) *norm[3]*, GLuint & *nf*, GLuint(*&) *face[3]*, bool *normalize* = false)

三角形分割された OBJ ファイルを読み込む (Elements 形式).

三角形分割された OBJ ファイルを読み込む.

引数

name	読み込む Wavefront OBJ ファイル名.
nv	読み込んだデータの頂点数.
pos	読み込んだデータの頂点座標.
norm	読み込んだデータの頂点法線.
face	読み込んだデータの三角形の頂点インデックス.
normalize	true なら読み込んだデータの大きさを正規化する.

戻り値

ファイルの読み込みに成功したら true.

引数

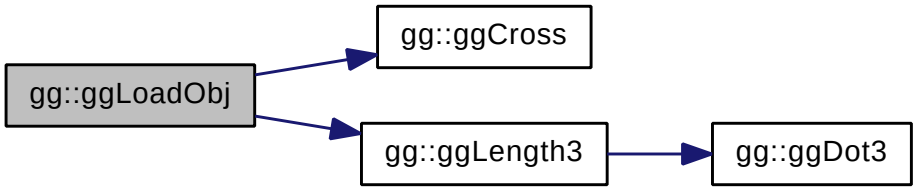
name	OBJ ファイル名.
nv	読み込んだデータの頂点数を格納する変数.
pos	頂点の位置のデータを格納したメモリのポインタを格納する変数.
norm	頂点の法線データの格納したメモリのポインタを格納する変数.
nf	読み込んだデータの面数を格納する変数.
face	面のデータを格納したメモリのポインタを格納する変数.
normalize	true ならサイズを正規化する.

戻り値

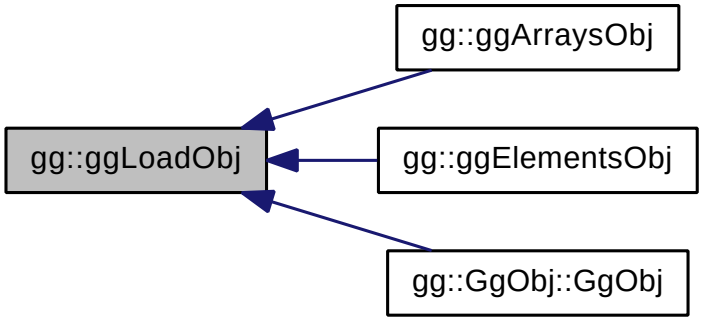
読み込みに成功したら true.

gg.cpp の 5792 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:



5.1.1.26 bool gg::ggLoadObj (const char * *name*, GLuint & *ng*, GLuint(*&) *group[2]*, GLfloat(*&) *amb[4]*, GLfloat(*&) *diff[4]*, GLfloat(*&) *spec[4]*, GLfloat *& *shi*, GLuint & *nv*, GLfloat(*&) *pos[3]*, GLfloat(*&) *norm[3]*, bool *normalize* = `false`)

三角形分割された OBJ ファイルと MTL ファイルを読み込む (Arrays 形式)

三角形分割された OBJ ファイルと MTL ファイルを読み込む.

引数

name	読み込む Wavefront OBJ ファイル名.
ng	読み込んだデータの面グループ数.
group	読み込んだデータの面グループの最初の面のインデックスと面数.
amb	読み込んだデータの面グループごとの環境光に対する反射係数.
diff	読み込んだデータの面グループごとの拡散反射係数.
spec	読み込んだデータの面グループごとの鏡面反射係数.
shi	読み込んだデータの面グループごとの輝き係数.
nv	読み込んだデータの頂点数.
pos	読み込んだデータの頂点座標.
norm	読み込んだデータの頂点法線.
normalize	true なら読み込んだデータの大きさを正規化する.

戻り値

ファイルの読み込みに成功したら true.

引数

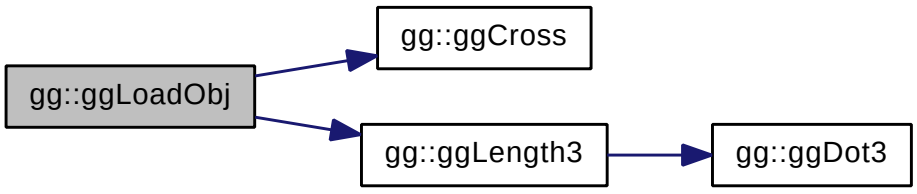
name	OBJ ファイル名.
ng	読み込んだデータの面のグループ数.
group	面の各グループの最初の面の番号と面の数.
amb	グループごとの環境光に対する反射係数.
diff	グループごとの拡散反射係数.
spec	グループごとの鏡面反射係数.
shi	グループごとの輝き係数.
nv	読み込んだデータの頂点数を格納する変数.
pos	頂点の位置のデータを格納したメモリのポインタを格納する変数.
norm	頂点の法線データの格納したメモリのポインタを格納する変数.
normalize	true ならサイズを正規化する.

戻り値

読み込みに成功したら true.

gg.cpp の 5994 行目に定義があります。

呼び出し関係図:



5.1.1.27 GLuint gg::ggLoadShader (const char * *vert*, const char * *frag* = NULL, const char * *geom* = NULL, GLint *nvarying* = 0, const char *const *varyings[]* = NULL)

シェーダのソースファイルを読み込んでプログラムオブジェクトを作成する.

引数

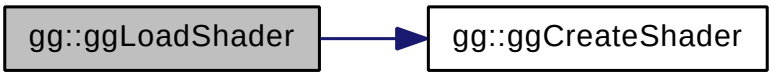
vert	バーテックスシェーダのソースファイル名.
frag	フラグメントシェーダのソースファイル名 (NULL なら不使用).
geom	ジオメトリシェーダのソースファイル名 (NULL なら不使用).
nvarying	フィードバックする varying 変数の数 (0 なら不使用).
varyings	フィードバックする varying 変数のリスト (NULL なら不使用).

戻り値

シェーダプログラムのプログラム名 (作成できなければ 0).

gg.cpp の 6728 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:

5.1.1.28 GLuint gg::ggLoadTexture (GLsizei *width*, GLsizei *height*, GLenum *internal*, GLenum *format* = GL_RGBA, const GLvoid * *image* = NULL)

テクスチャメモリを確保して画像データをテクスチャとして読み込む.

テクスチャメモリを確保して画像を読み込む.

引数

width	テクスチャとして読み込むデータ image の横の画素数.
height	テクスチャとして読み込むデータ image の縦の画素数.
internal	glTexImage2D() に指定するテクスチャの内部フォーマット.
format	glTexImage2D() に指定するデータ image のフォーマット.
image	テクスチャとして読み込むデータ. NULL ならテクスチャメモリの確保のみ.
width	画像の幅.
height	画像の高さ.
internal	テクスチャの内部フォーマット.
format	画像データのフォーマット.
image	画像データ.

戻り値

テクスチャオブジェクト名.

gg.cpp の 5534 行目に定義があります。

被呼び出し関係図:

5.1.1.29 GLubyte * gg::ggLoadTga (const char * *name*, GLsizei * *width*, GLsizei * *height*, GLenum * *format*)

TGA ファイル (8/16/24/32bit) をメモリに読み込む.

TGA ファイル (8/16/24/32bit) を読み込む.

引数

name	読み込むファイル名.
width	読み込んだファイルの横の画素数.
height	読み込んだファイルの縦の画素数.
format	読み込んだファイルの書式. GL_RED, G_RG, GL_BGR, G_BGRA.

戻り値

読み込みに成功すれば読み込んだデータのポインタ, 失敗すれば NULL.

引数

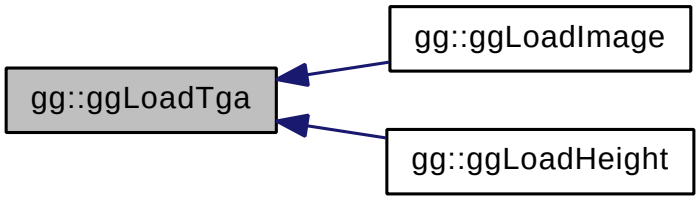
name	読み込むファイル名.
width	読み込んだファイルの幅.
height	読み込んだファイルの高さ.
format	読み込んだファイルのフォーマット.

戻り値

読み込んだ画像データのポインタ (要 delete, 読み込めなければ NULL)

gg.cpp の 5409 行目に定義があります。

被呼び出し関係図:



5.1.1.30 GgMatrix gg::ggLookat (GLfloat *ex*, GLfloat *ey*, GLfloat *ez*, GLfloat *tx*, GLfloat *ty*, GLfloat *tz*, GLfloat *ux*, GLfloat *uy*, GLfloat *uz*) [inline]

ビュー変換行列を返す.

引数

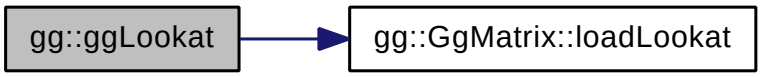
ex	視点の位置の x 座標値.
ey	視点の位置の y 座標値.
ez	視点の位置の z 座標値.
tx	目標点の位置の x 座標値.
ty	目標点の位置の y 座標値.
tz	目標点の位置の z 座標値.
ux	上方向のベクトルの x 成分.
uy	上方向のベクトルの y 成分.
uz	上方向のベクトルの z 成分.

戻り値

求めたビュー変換行列.

gg.h の 3577 行目に定義があります。

呼び出し関係図:



5.1.1.31 GgMatrix gg::ggLookat (const GLfloat * e, const GLfloat * t, const GLfloat * u) [inline]

ビュー変換行列を返す.

引数

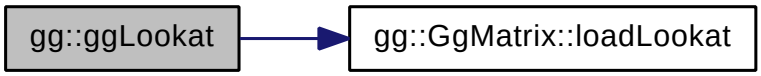
e	視点の位置の配列変数.
t	目標点の位置の配列変数.
u	上方向のベクトルのの配列変数.

戻り値

求めたビュー変換行列.

gg.h の 3592 行目に定義があります。

呼び出し関係図:



5.1.1.32 GgQuaternion gg::ggMatrixQuaternion (const GLfloat * a) [inline]

回転の変換行列 m を表す四元数を返す.

引数

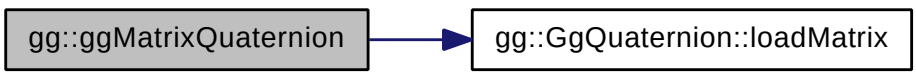
m	GLfloat 型の 16 要素の配列.

戻り値

m による回転の変換に相当する四元数.

gg.h の 4455 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:

5.1.1.33 GgQuaternion gg::ggMatrixQuaternion (const GgMatrix & *m*) [inline]

回転の変換行列 m を表す四元数を返す.

引数

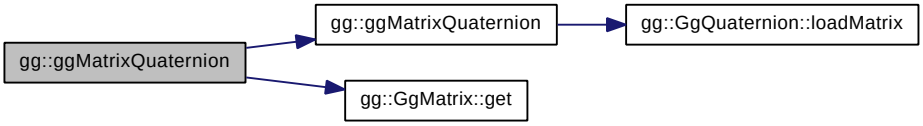
m	GgMatrix 型の変換行列.

戻り値

m による回転の変換に相当する四元数.

gg.h の 4464 行目に定義があります。

呼び出し関係図:



5.1.1.34 GLfloat gg::ggNorm (const GgQuaternion & *q*) [inline]

四元数のノルムを返す.

引数

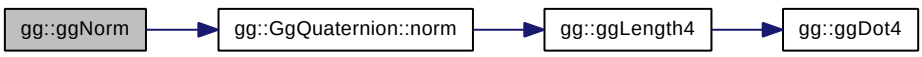
q	GgQuaternion 型の四元数.

戻り値

四元数 q のノルム.

gg.h の 4583 行目に定義があります。

呼び出し関係図:



5.1.1.35 GgMatrix gg::ggNormal (const GgMatrix & *m*) [inline]

法線変換行列を返す.

引数

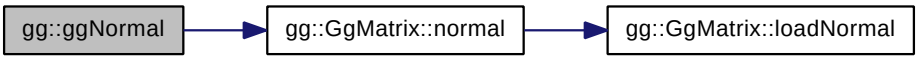
m	元の変換行列.

戻り値

m の法線変換行列.

gg.h の 3666 行目に定義があります。

呼び出し関係図:



5.1.1.36 GgQuaternion gg::ggNormalize (const GgQuaternion & *q*) [inline]

正規化した四元数を返す.

引数

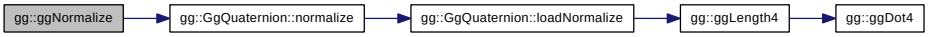
q	GgQuaternion 型の四元数.

戻り値

四元数 q を正規化した四元数.

gg.h の 4591 行目に定義があります。

呼び出し関係図:



5.1.1.37 GgMatrix gg::ggOrthogonal (GLfloat *left*, GLfloat *right*, GLfloat *bottom*, GLfloat *top*, GLfloat *zNear*, GLfloat *zFar*) [inline]

直交投影変換行列を返す.

引数

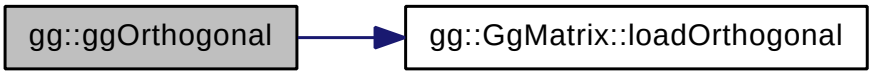
left	ウィンドウの左端の位置.
right	ウィンドウの右端の位置.
bottom	ウィンドウの下端の位置.
top	ウィンドウの上端の位置.
zNear	視点から前方面までの位置.
zFar	視点から後方面までの位置.

戻り値

求めた直交投影変換行列.

gg.h の 3610 行目に定義があります。

呼び出し関係図:



5.1.1.38 GgMatrix gg::ggPerspective (GLfloat *fovy*, GLfloat *aspect*, GLfloat *zNear*, GLfloat *zFar*) [inline]

画角を指定して透視投影変換行列を返す.

引数

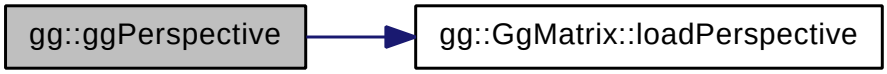
fovy	y 方向の画角.
aspect	縦横比.
zNear	視点から前方面までの位置.
zFar	視点から後方面までの位置.

戻り値

求めた透視投影変換行列.

gg.h の 3640 行目に定義があります。

呼び出し関係図:



5.1.1.39 gg::GgPoints * gg::ggPointsCube (GLuint *nv*, GLfloat *length* = 1.0f, GLfloat *cx* = 0.0f, GLfloat *cy* = 0.0f, GLfloat *cz* = 0.0f)

点群を立方体状に生成する.

引数

nv	生成する点の数.
length	点群を生成する立方体の一辺の長さ.
cx	点群の中心の x 座標.
cy	点群の中心の y 座標.
cz	点群の中心の z 座標.
nv	生成する点の数.
cx	点群の中心の x 座標.
cy	点群の中心の y 座標.
cz	点群の中心の z 座標.
length	点群を生成する立方体の一辺の長さ.

gg.cpp の 7557 行目に定義があります。

5.1.1.40 gg::GgPoints * gg::ggPointsSphere (GLuint *nv*, GLfloat *radius* = 0.5f, GLfloat *cx* = 0.0f, GLfloat *cy* = 0.0f, GLfloat *cz* = 0.0f)

点群を球状に生成する.

引数

nv	生成する点の数.
radius	点群を生成する半径.
cx	点群の中心の x 座標.
cy	点群の中心の y 座標.
cz	点群の中心の z 座標.
nv	生成する点の数.
cx	点群の中心の x 座標.
cy	点群の中心の y 座標.
cz	点群の中心の z 座標.
radius	点群を生成する半径.

gg.cpp の 7592 行目に定義があります。

5.1.1.41 GgQuaternion gg::ggQuaternion (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *w*) [inline]

四元数を返す

引数

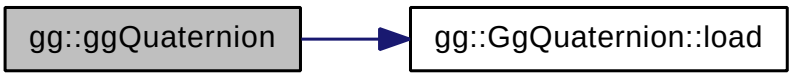
x	四元数の x 要素.
y	四元数の y 要素.
z	四元数の z 要素.
w	四元数の w 要素.

戻り値

四元数.

gg.h の 4430 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:

5.1.1.42 GgQuaternion gg::ggQuaternion (const GLfloat * *a*) [inline]

四元数を返す

引数

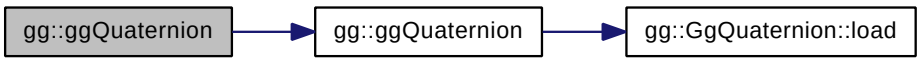
a	GLfloat 型の GLfloat 型の 4 要素の配列に格納した四元数.

戻り値

四元数.

gg.h の 4439 行目に定義があります。

呼び出し関係図:



5.1.1.43 GgMatrix gg::ggQuaternionMatrix (const GgQuaternion & *q*) [inline]

四元数 q の回転の変換行列を返す.

引数

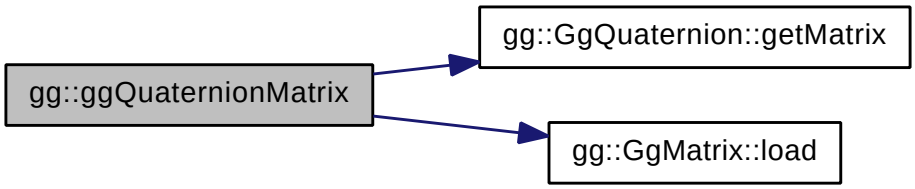
q	元の四元数.

戻り値

四元数 q が表す回転に相当する GgMatrix 型の変換行列.

gg.h の 4472 行目に定義があります。

呼び出し関係図:



5.1.1.44 GgMatrix gg::ggQuaternionTransposeMatrix (const GgQuaternion & *q*) [inline]

四元数 q の回転の転置した変換行列を返す.

引数

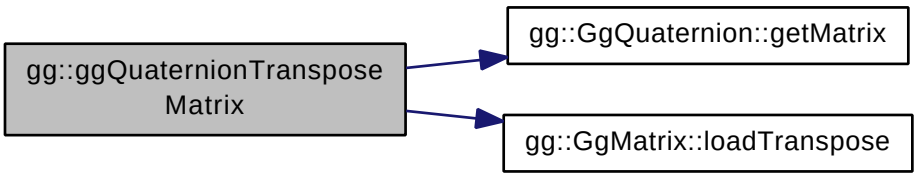
q	元の四元数.

戻り値

四元数 q が表す回転に相当する転置した GgMatrix 型の変換行列.

gg.h の 4483 行目に定義があります。

呼び出し関係図:



5.1.1.45 gg::GgTriangles * gg::ggRectangle (GLfloat *width* = 1.0f, GLfloat *height* = 1.0f)

矩形状に 2 枚の三角形を生成する.

引数

width	矩形の幅.
height	矩形の高さ.

gg.cpp の 7632 行目に定義があります。

5.1.1.46 GgMatrix gg::ggRotate (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *a*) [inline]

(x, y, z) 方向のベクトルを軸とする回転の変換行列を乗じた結果を返す.

引数

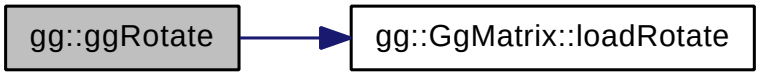
x	回転軸の x 成分.
y	回転軸の y 成分.
z	回転軸の z 成分.
a	回転角.

戻り値

(x, y, z) を軸にさらに a 回転する変換行列.

gg.h の 3541 行目に定義があります。

呼び出し関係図:



5.1.1.47 GgMatrix gg::ggRotate (const GLfloat * *r*, GLfloat *a*) [inline]

r 方向のベクトルを軸とする回転の変換行列を乗じた結果を返す.

引数

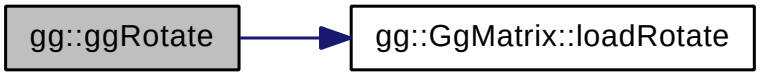
r	回転軸のベクトルを表す GLfloat 型の 3 要素の配列.
a	回転角.

戻り値

r を軸に a だけ回転する変換行列.

gg.h の 3551 行目に定義があります。

呼び出し関係図:



5.1.1.48 GgMatrix gg::ggRotate (const GLfloat * r) [inline]

r 方向のベクトルを軸とする回転の変換行列を乗じた結果を返す.

引数

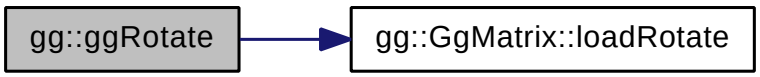
r	回転軸のベクトルと回転角を表す GLfloat 型の 4 要素の配列.

戻り値

(r[0], r[1], r[2]) を軸に r[3] だけ回転する変換行列.

gg.h の 3560 行目に定義があります。

呼び出し関係図:



5.1.1.49 GgQuaternion gg::ggRotateQuaternion (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *a*) [inline]

(x, y, z) を軸として角度 a 回転する四元数を返す.

引数

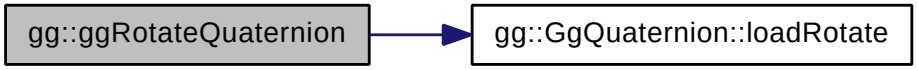
x	軸ベクトルの x 成分.
y	軸ベクトルの y 成分.
z	軸ベクトルの z 成分.
a	回転角.

戻り値

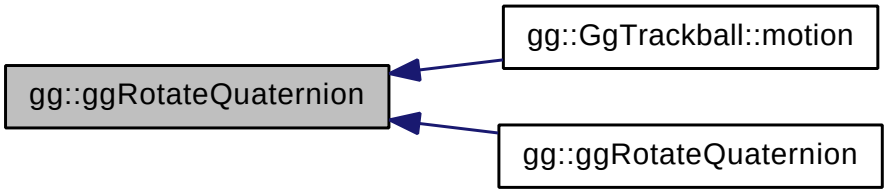
回転を表す四元数.

gg.h の 4497 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:



5.1.1.50 GgQuaternion gg::ggRotateQuaternion (const GLfloat * *v*, GLfloat *a*) [inline]

(v[0], v[1], v[2]) を軸として角度 a 回転する四元数を返す.

引数

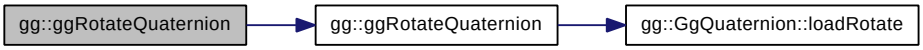
v	軸ベクトルを表す GLfloat 型の 3 要素の配列.
a	回転角.

戻り値

回転を表す四元数.

gg.h の 4507 行目に定義があります。

呼び出し関係図:



5.1.1.51 GgQuaternion gg::ggRotateQuaternion (const GLfloat * *v*) [inline]

(v[0], v[1], v[2]) を軸として角度 v[3] 回転する四元数を返す.

引数

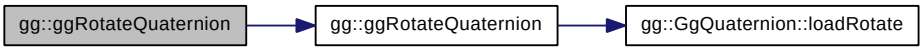
v	軸ベクトルを表す GLfloat 型の 4 要素の配列.

戻り値

回転を表す四元数.

gg.h の 4515 行目に定義があります。

呼び出し関係図:



5.1.1.52 GgMatrix gg::ggRotateX (GLfloat *a*) [inline]

x 軸中心の回転の変換行列を返す.

引数

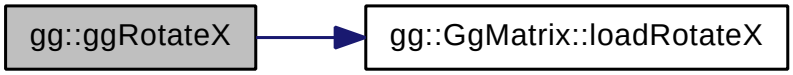
a	回転角.

戻り値

x 軸中心に a だけ回転する変換行列.

gg.h の 3511 行目に定義があります。

呼び出し関係図:



5.1.1.53 GgMatrix gg::ggRotateY (GLfloat *a*) [inline]

y 軸中心の回転の変換行列を返す.

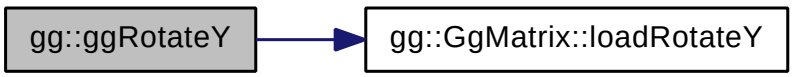
引数

a	回転角.

戻り値

y 軸中心に a だけ回転する変換行列.

gg.h の 3520 行目に定義があります。
呼び出し関係図:



5.1.1.54 GgMatrix gg::ggRotateZ (GLfloat *a*) [inline]

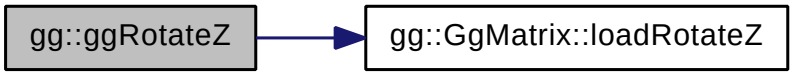
z 軸中心の回転の変換行列を返す.
引数

a	回転角.

戻り値

z 軸中心に a だけ回転する変換行列.

gg.h の 3529 行目に定義があります。
呼び出し関係図:



5.1.1.55 bool gg::ggSaveColor (const char * *name*)

カラーバッファの内容を TGA ファイルに保存する.
引数

name	保存するファイル名.

戻り値

保存に成功すれば true, 失敗すれば false.

引数

name	保存するファイル名.

戻り値

保存に成功したら true.

gg.cpp の 5352 行目に定義があります。

呼び出し関係図:

5.1.1.56 bool gg::ggSaveDepth (const char * *name*)

デプスバッファの内容を TGA ファイルに保存する.

引数

name	保存するファイル名.

戻り値

保存に成功すれば true, 失敗すれば false.

引数

name	保存するファイル名.

戻り値

保存に成功したら true.

gg.cpp の 5379 行目に定義があります。

呼び出し関係図:

5.1.1.57 bool gg::ggSaveTga (GLsizei *sx*, GLsizei *sy*, unsigned int *depth*, const void * *buffer*, const char * *name*)

配列の内容を TGA ファイルに保存する.

配列に格納された画像の内容を TGA ファイルに保存する.

引数

sx	画像の横の画素数.
sy	画像の縦の画素数.
depth	1 画素のバイト数.

buffer	画像データを格納した配列.
name	保存するファイル名.

戻り値

保存に成功すれば true, 失敗すれば false.

引数

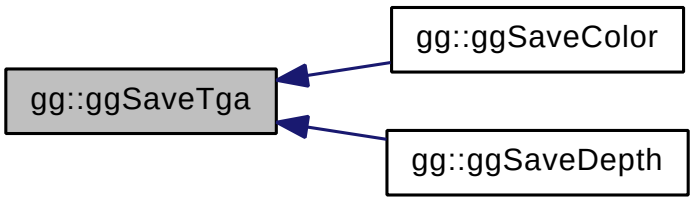
sx	画像の幅.
sy	画像の高さ.
depth	画像の 1 画素のバイト数.
buffer	画像データ.
name	ファイル名.

戻り値

保存に成功したら true.

gg.cpp の 5260 行目に定義があります。

被呼び出し関係図:



5.1.1.58 GgMatrix gg::ggScale (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *w* = `1.0f`) [inline]

拡大縮小の変換行列を返す.

引数

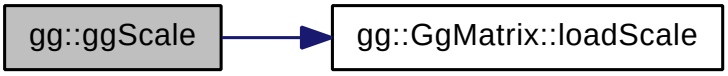
x	x 方向の拡大率.
y	y 方向の拡大率.
z	z 方向の拡大率.
w	拡大率のスケールファクタ (= 1.0f).

戻り値

拡大縮小の変換行列.

gg.h の 3493 行目に定義があります。

呼び出し関係図:



5.1.1.59 GgMatrix gg::ggScale (const GLfloat * s) [inline]

拡大縮小の変換行列を返す.

引数

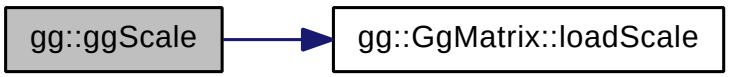
s	拡大率の GLfloat 型の配列 (x, y, z).

戻り値

拡大縮小の変換行列.

gg.h の 3502 行目に定義があります。

呼び出し関係図:



5.1.1.60 GgQuaternion gg::ggSlerp (const GLfloat * *a*, const GLfloat * *b*, GLfloat *t*) [inline]

二つの四元数の球面線形補間の結果を返す.

引数

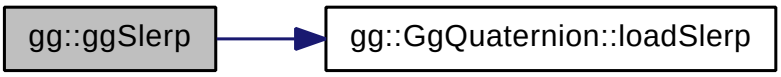
a	四元数を格納した GLfloat 型の 4 要素の配列.
b	四元数を格納した GLfloat 型の 4 要素の配列.
t	補間パラメータ.

戻り値

a, b を t で内分した四元数.

gg.h の 4544 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:

5.1.1.61 GgQuaternion gg::ggSlerp (const GgQuaternion & *q*, const GgQuaternion & *r*, GLfloat *t*) [inline]

二つの四元数の球面線形補間の結果を返す.

引数

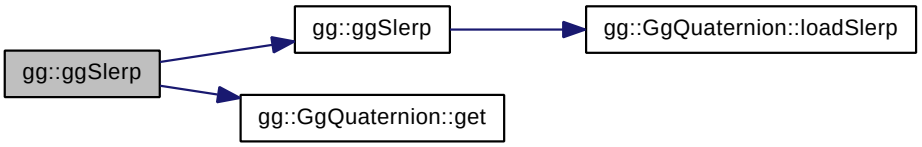
q	GgQuaternion 型の四元数.
r	GgQuaternion 型の四元数.
t	補間パラメータ.

戻り値

q, r を t で内分した四元数.

gg.h の 4555 行目に定義があります。

呼び出し関係図:



5.1.1.62 GgQuaternion gg::ggSlerp (const GgQuaternion & *q*, const GLfloat * *a*, GLfloat *t*) [inline]

二つの四元数の球面線形補間の結果を返す.

引数

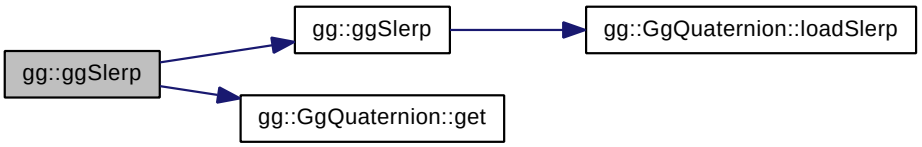
q	GgQuaternion 型の四元数.
a	四元数を格納した GLfloat 型の 4 要素の配列.
t	補間パラメータ.

戻り値

q, a を t で内分した四元数.

gg.h の 4565 行目に定義があります。

呼び出し関係図:



5.1.1.63 GgQuaternion gg::ggSlerp (const GLfloat * *a*, const GgQuaternion & *q*, GLfloat *t*) [inline]

二つの四元数の球面線形補間の結果を返す.

引数

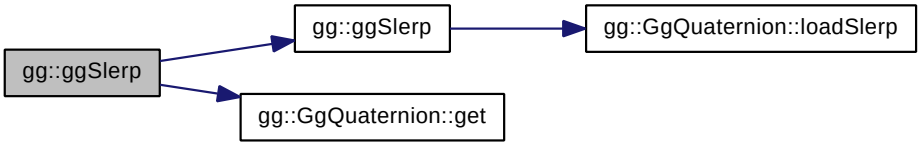
a	四元数を格納した GLfloat 型の 4 要素の配列.
q	GgQuaternion 型の四元数.
t	補間パラメータ.

戻り値

a, q を t で内分した四元数.

gg.h の 4575 行目に定義があります。

呼び出し関係図:



5.1.1.64 GgMatrix gg::ggTranslate (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *w* = 1.0f) [inline]

平行移動の変換行列を返す.

引数

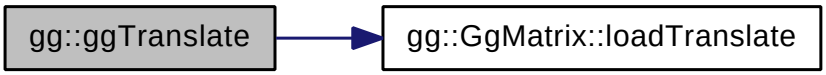
x	x 方向の移動量.
y	y 方向の移動量.
z	z 方向の移動量.
w	移動量のスケールファクタ (= 1.0f).

戻り値

平行移動の変換行列.

gg.h の 3472 行目に定義があります。

呼び出し関係図:



5.1.1.65 GgMatrix gg::ggTranslate (const GLfloat * *t*) [inline]

平行移動の変換行列を返す.

引数

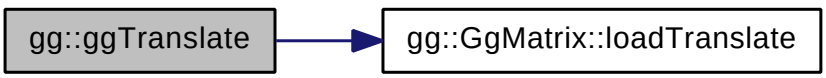
t	移動量の GLfloat 型の配列 (x, y, z).

戻り値

平行移動の変換行列

gg.h の 3481 行目に定義があります。

呼び出し関係図:



5.1.1.66 GgMatrix gg::ggTranspose (const GgMatrix & *m*) [inline]

転置行列を返す.

引数

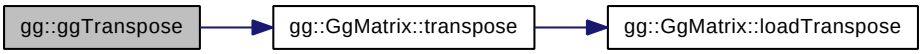
m	元の変換行列.

戻り値

m の転置行列.

gg.h の 3650 行目に定義があります。

呼び出し関係図:



Chapter 6

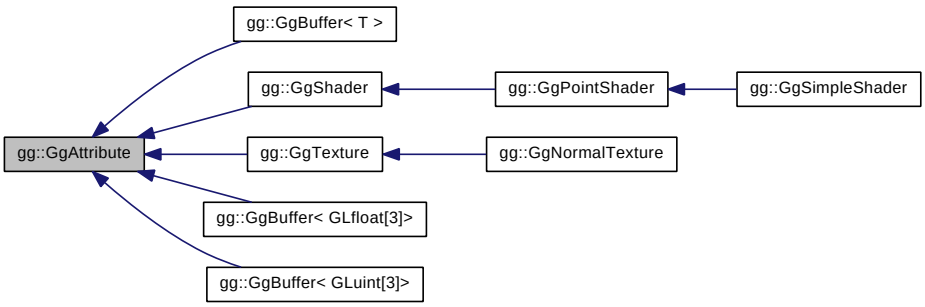
クラス詳解

6.1 gg::GgAttribute クラス

属性データ.

`#include <gg.h>`

gg::GgAttribute の継承関係図



公開メンバ関数

- virtual ~GgAttribute ()

  デストラクタ.
- GgAttribute ()

  コンストラクタ.
- GgAttribute (const GgAttribute &o)

  コピーコンストラクタ.
- GgAttribute & operator= (const GgAttribute &o)

限定公開メンバ関数

- bool unique () const

  唯一のオブジェクトかどうか調べる.
- bool reset ()

  参照カウンタの新規作成.

6.1.1 詳解

属性データ.

テクスチャとシェーダの基底クラス. インスタンスは複数のオブジェクトから参照されることを想定する. そのためこのクラスでは参照カウントを管理する.

gg.h の 4725 行目に定義があります。

6.1.2 構築子と解体子

6.1.2.1 virtual gg::GgAttribute::~GgAttribute ( ) [inline], [virtual]

デストラクタ.

gg.h の 4759 行目に定義があります。

6.1.2.2 gg::GgAttribute::GgAttribute ( ) [inline]

コンストラクタ.

gg.h の 4766 行目に定義があります。

6.1.2.3 gg::GgAttribute::GgAttribute (const GgAttribute & *o*) [inline]

コピーコンストラクタ.

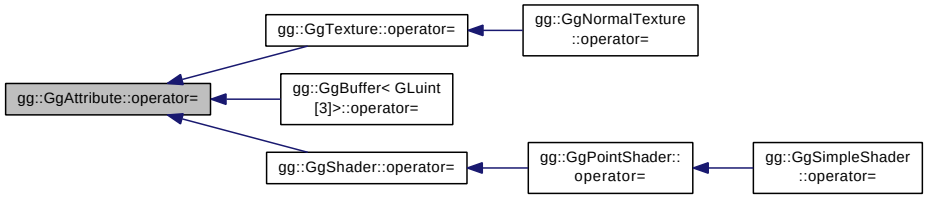
gg.h の 4770 行目に定義があります。

6.1.3 関数詳解

6.1.3.1 GgAttribute& gg::GgAttribute::operator= (const GgAttribute & *o*) [inline]

gg.h の 4777 行目に定義があります。

被呼び出し関係図:



6.1.3.2 bool gg::GgAttribute::reset ( ) [inline], [protected]

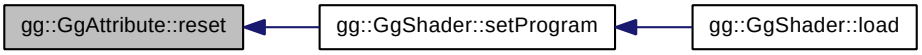
参照カウンタの新規作成.

戻り値

唯一のオブジェクトなら真.

gg.h の 4741 行目に定義があります。

被呼び出し関係図:



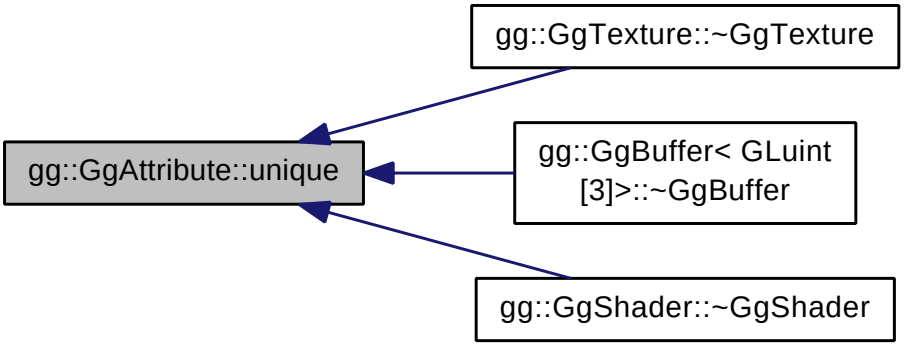
6.1.3.3 bool gg::GgAttribute::unique ( ) const [inline], [protected]

唯一のオブジェクトかどうか調べる.

戻り値

唯一のオブジェクトなら真.

gg.h の 4734 行目に定義があります。
被呼び出し関係図:



このクラス詳解は次のファイルから抽出されました:

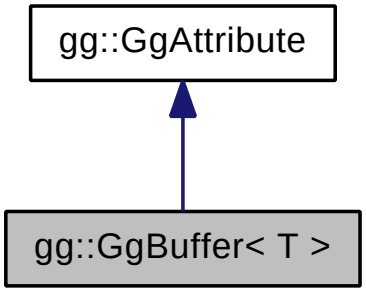
- gg.h

6.2 gg::GgBuffer< T > クラステンプレート

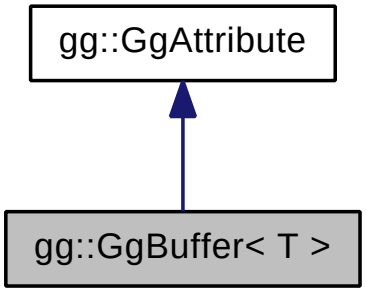
バッファオブジェクト.

`#include <gg.h>`

gg::GgBuffer< T > の継承関係図



gg::GgBuffer< T > 連携図



公開メンバ関数

- virtual ~GgBuffer ()
  デストラクタ.
- GgBuffer ()
  コンストラクタ.
- GgBuffer (GLenum target, GLuint number, const T *data, GLenum usage=GL_STATIC_DRAW)
  コンストラクタ.
- GgBuffer (const GgBuffer< T > &o)
  コピーコンストラクタ.
- GgBuffer< T > & operator= (const GgBuffer< T > &o)
- void send (GLuint number, const T *data, GLuint offset=0)
  すでに確保したバッファオブジェクトにデータを転送する.
- void load (GLenum target, GLuint number, const T *data, GLenum usage=GL_STATIC_DRAW)
  バッファオブジェクトを確保してデータを格納する.
- void copy (GLuint buf)
  別のバッファオブジェクトからデータを複写する.
- GLuint buf () const
  バッファオブジェクト名を取り出す.
- GLuint num () const
  バッファオブジェクトが保持するデータの数を取り出す.

その他の継承メンバ

6.2.1 詳解

template<typename T>class gg::GgBuffer< T >

バッファオブジェクト.

頂点／インデックスを格納する頂点バッファオブジェクトの基底クラス.

gg.h の 4909 行目に定義があります。

6.2.2 構築子と解体子

6.2.2.1 template<typename T> virtual gg::GgBuffer< T >::~GgBuffer ( ) [inline], [virtual]

デストラクタ.

gg.h の 4924 行目に定義があります。

6.2.2.2 template<typename T> gg::GgBuffer< T >::GgBuffer ( ) [inline]

コンストラクタ.

gg.h の 4935 行目に定義があります。

6.2.2.3 template<typename T> gg::GgBuffer< T >::GgBuffer (GLenum *target*, GLuint *number*, const T * *data*, GLenum *usage* = GL_STATIC_DRAW) [inline]

コンストラクタ.

引数

target	バッファオブジェクトのターゲット.
number	データの数.
data	データが格納されている領域の先頭のポインタ (NULL ならデータを転送しない).
usage	バッファオブジェクトの使い方.

gg.h の 4946 行目に定義があります。

6.2.2.4 template<typename T> gg::GgBuffer< T >::GgBuffer (const GgBuffer< T > & *o*) [inline]

コピーコンストラクタ.

gg.h の 4953 行目に定義があります。

6.2.3 関数詳解

6.2.3.1 template<typename T> GLuint gg::GgBuffer< T >::buf ( ) const [inline]

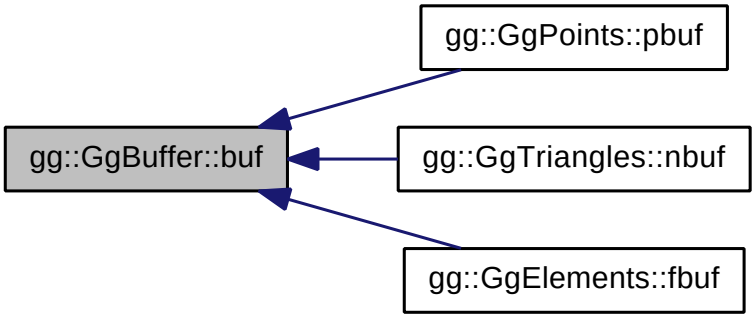
バッファオブジェクト名を取り出す.

戻り値

バッファオブジェクト名.

gg.h の 5005 行目に定義があります。

被呼び出し関係図:



6.2.3.2 template<typename T> void gg::GgBuffer< T >::copy (GLuint *buf*) [inline]

別のバッファオブジェクトからデータを複写する.

引数

buf	コピー元のバッファオブジェクト名.

gg.h の 4994 行目に定義があります。

6.2.3.3 template<typename T> void gg::GgBuffer< T >::load (GLenum *target*, GLuint *number*, const T * *data*, GLenum *usage* = GL_STATIC_DRAW) [inline]

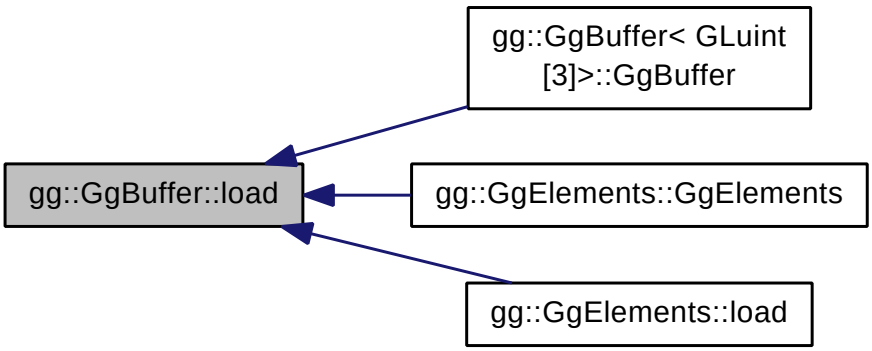
バッファオブジェクトを確保してデータを格納する.

引数

target	バッファオブジェクトのターゲット.
number	データの数.
data	データが格納されている領域の先頭のポインタ.
usage	バッファオブジェクトの使い方.

gg.h の 4984 行目に定義があります。

被呼び出し関係図:



6.2.3.4 template<typename T> GLuint gg::GgBuffer< T >::num ( ) const [inline]

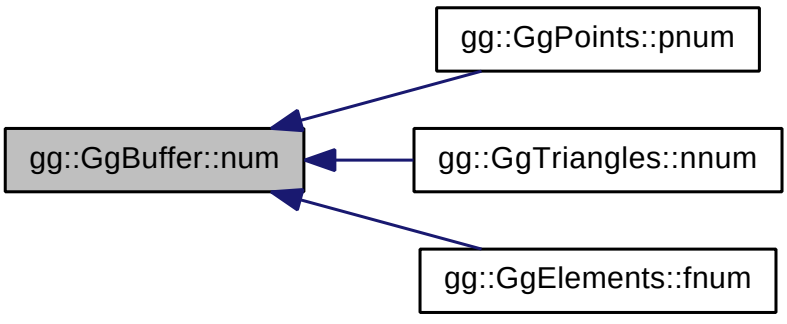
バッファオブジェクトが保持するデータの数を取り出す.

戻り値

保持するデータの数.

gg.h の 5012 行目に定義があります。

被呼び出し関係図:



6.2.3.5 template<typename T> GgBuffer<T>& gg::GgBuffer< T >::operator= (const GgBuffer< T > & *o*) [inline]

gg.h の 4957 行目に定義があります。

6.2.3.6 template<typename T> void gg::GgBuffer< T >::send (GLuint *number*, const T * *data*, GLuint *offset* = 0) [inline]

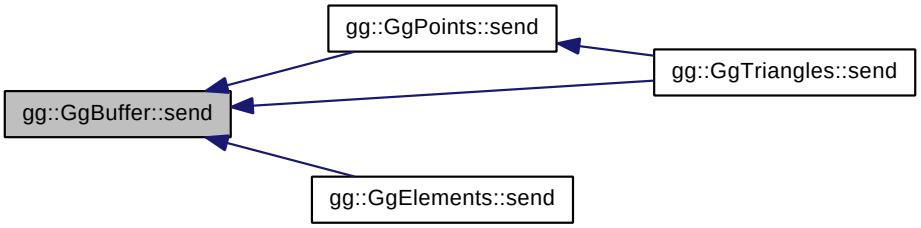
すでに確保したバッファオブジェクトにデータを転送する.

引数

number	転送するデータの数 (0 ならバッファ全体).
data	転送元のデータが格納されてている領域の先頭のポインタ.
offset	転送先のバッファオブジェクトの先頭の要素番号.

gg.h の 4973 行目に定義があります。

被呼び出し関係図:



このクラス詳解は次のファイルから抽出されました:

- gg.h

6.3 gg::GgCounter クラス

参照カウンタ.

`#include <gg.h>`

フレンド

- class GgAttribute

6.3.1 詳解

参照カウンタ.

複数の属性データ間で共有されるリソースの確保と解放を管理する

gg.h の 4702 行目に定義があります。

6.3.2 フレンドと関連関数の詳解

6.3.2.1 friend class GgAttribute [friend]

gg.h の 4715 行目に定義があります。

このクラス詳解は次のファイルから抽出されました:

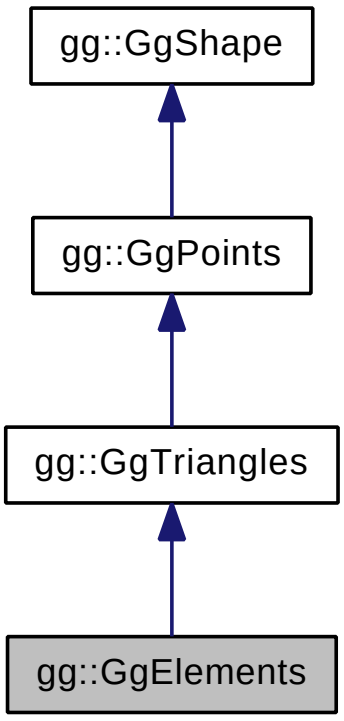
- gg.h

6.4 gg::GgElements クラス

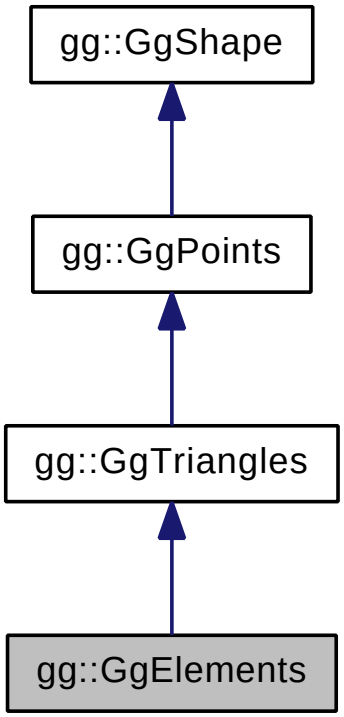
三角形で表した形状データ (Elements 形式).

`#include <gg.h>`

gg::GgElements の継承関係図



gg::GgElements 連携図



公開メンバ関数

- virtual ~GgElements ()

  デストラクタ.
- GgElements (GLenum mode=GL_TRIANGLES)

  コンストラクタ.
- GgElements (GLuint nv, const GLfloat(∗pos)[3], const GLfloat(∗norm)[3], GLuint nf, const GLuint(∗face)[3], GLenum mode=GL_TRIANGLES, GLenum usage=GL_STATIC_DRAW)

  コンストラクタ.
- GgElements (const GgElements &o)

  コピーコンストラクタ.
- GgElements & operator= (const GgElements &o)
- void send (GLuint nf, const GLuint(∗face)[3], GLuint offset=0)

  既存のバッファオブジェクトに三角形の頂点インデックスデータを転送する.
- void load (GLuint nv, const GLfloat(∗pos)[3], const GLfloat(∗norm)[3], GLuint nf, const GLuint(∗face)[3], GLenum usage=GL_STATIC_DRAW)

  バッファオブジェクトを確保して頂点の位置データと法線データと三角形の頂点インデックスデータを格納する.
- GLuint fbuf () const

  三角形の頂点インデックスデータを格納した頂点バッファオブジェクト名を取り出す.
- GLuint fnum () const

  データの数を取り出す.
- virtual void draw (GLint first=0, GLsizei count=0) const

  三角形を描画する手続き.

6.4.1 詳解

三角形で表した形状データ (Elements 形式).

gg.h の 5296 行目に定義があります。

6.4.2 構築子と解体子

6.4.2.1 virtual gg::GgElements::~GgElements ( ) [inline], [virtual]

デストラクタ.

gg.h の 5305 行目に定義があります。

6.4.2.2 gg::GgElements::GgElements (GLenum *mode* = GL_TRIANGLES) [inline]

コンストラクタ.

引数

mode	描画する基本図形の種類.

gg.h の 5309 行目に定義があります。

6.4.2.3 gg::GgElements::GgElements (GLuint *nv*, const GLfloat(*) *pos[3]*, const GLfloat(*) *norm[3]*, GLuint *nf*, const GLuint(*) *face[3]*, GLenum *mode* = GL_TRIANGLES, GLenum *usage* = GL_STATIC_DRAW) [inline]

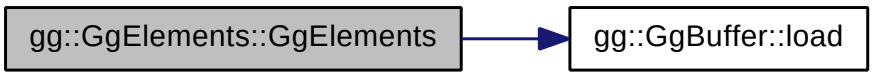
コンストラクタ.

引数

nv	頂点数.
pos	この図形の頂点の位置のデータの配列 (NULL ならデータを転送しない).
norm	この図形の頂点の法線のデータの配列 (NULL ならデータを転送しない).
nf	三角形数.
face	三角形の頂点インデックス.
mode	描画する基本図形の種類.
usage	バッファオブジェクトの使い方.

gg.h の 5320 行目に定義があります。

呼び出し関係図:



6.4.2.4 gg::GgElements::GgElements (const GgElements & *o*) [inline]

コピーコンストラクタ.

gg.h の 5328 行目に定義があります。

6.4.3 関数詳解

6.4.3.1 void gg::GgElements::draw (GLint *first* = 0, GLsizei *count* = 0) const [virtual]

三角形を描画する手続き.

gg::GgPointsを再実装しています。

gg.cpp の 7538 行目に定義があります。

6.4.3.2 GLuint gg::GgElements::fbuf ( ) const [inline]

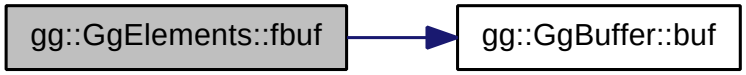
三角形の頂点インデックスデータを格納した頂点バッファオブジェクト名を取り出す.

戻り値

この図形の三角形の頂点インデックスデータを格納した頂点バッファオブジェクト名.

gg.h の 5367 行目に定義があります。

呼び出し関係図:



6.4.3.3 GLuint gg::GgElements::fnum ( ) const [inline]

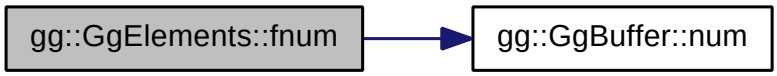
データの数を取り出す.

戻り値

この図形の三角形数.

gg.h の 5374 行目に定義があります。

呼び出し関係図:



6.4.3.4 void gg::GgElements::load (GLuint *nv*, const GLfloat(*) *pos[3]*, const GLfloat(*) *norm[3]*, GLuint *nf*, const GLuint(*) *face[3]*, GLenum *usage* = GL_STATIC_DRAW) [inline]

バッファオブジェクトを確保して頂点の位置データと法線データと三角形の頂点インデックスデータを格納する.

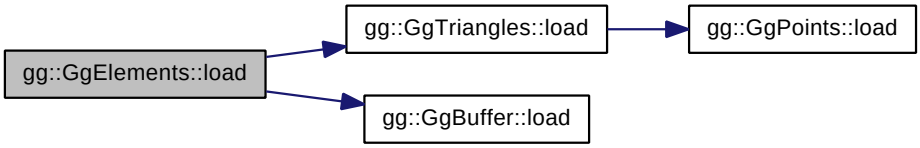
引数

nv	頂点のデータの数 (頂点数).
pos	頂点の位置データが格納されてている領域の先頭のポインタ.
norm	頂点の法線データが格納されてている領域の先頭のポインタ.

nf	三角形数.
face	三角形の頂点インデックスデータ.
usage	バッファオブジェクトの使い方.

gg.h の 5358 行目に定義があります。

呼び出し関係図:



6.4.3.5 GgElements& gg::GgElements::operator= (const GgElements & *o*) [inline]

gg.h の 5332 行目に定義があります。

呼び出し関係図:

6.4.3.6 void gg::GgElements::send (GLuint *nf*, const GLuint(*) *face[3]*, GLuint *offset* = 0) [inline]

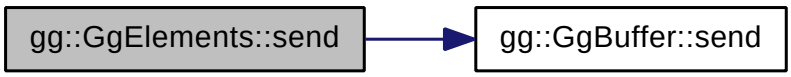
既存のバッファオブジェクトに三角形の頂点インデックスデータを転送する.

引数

nf	三角形数.
face	三角形の頂点インデックスデータ.
offset	転送先のバッファオブジェクトの先頭の要素番号.

gg.h の 5346 行目に定義があります。

呼び出し関係図:



このクラス詳解は次のファイルから抽出されました:

- gg.h
- gg.cpp

6.5 gg::GgMatrix クラス

変換行列.

`#include <gg.h>`

公開メンバ関数

- ~GgMatrix ()
  デストラクタ.
- GgMatrix ()
  コンストラクタ.
- GgMatrix (const GLfloat *a)
  コンストラクタ.
- GgMatrix (const GgMatrix &m)
  コピーコンストラクタ.
- GgMatrix & load (const GLfloat *a)
  配列変数の値を格納する.
- GgMatrix & load (const GgMatrix &m)
  別の変換行列の値を格納する.
- GgMatrix & loadAdd (const GLfloat *a)
  変換行列に配列に格納した変換行列を加算した結果を格納する.
- GgMatrix & loadAdd (const GgMatrix &m)
  変換行列に別の変換行列を加算した結果を格納する.
- GgMatrix & loadSubtract (const GLfloat *a)
  変換行列から配列に格納した変換行列を減算した結果を格納する.
- GgMatrix & loadSubtract (const GgMatrix &m)
  変換行列から別の変換行列を減算した結果を格納する.
- GgMatrix & loadMultiply (const GLfloat *a)
  変換行列に配列に格納した変換行列を乗算した結果を格納する.
- GgMatrix & loadMultiply (const GgMatrix &m)
  変換行列に別の変換行列を乗算した結果を格納する.
- GgMatrix & loadDivide (const GLfloat *a)
  変換行列を配列に格納した変換行列で除算した結果を格納する.
- GgMatrix & loadDivide (const GgMatrix &m)
  変換行列を別の変換行列で除算した結果を格納する.
- GgMatrix add (const GLfloat *a) const
  変換行列に配列に格納した変換行列を加算した値を返す.
- GgMatrix add (const GgMatrix &m) const
  変換行列に別の変換行列を加算した値を返す.
- GgMatrix subtract (const GLfloat *a) const
  変換行列から配列に格納した変換行列を減算した値を返す.
- GgMatrix subtract (const GgMatrix &m) const
  変換行列から別の変換行列を減算した値を返す.
- GgMatrix multiply (const GLfloat *a) const
  変換行列に配列に格納した変換行列を乗算した値を返す.
- GgMatrix multiply (const GgMatrix &m) const
  変換行列に別の変換行列を乗算した値を返す.
- GgMatrix divide (const GLfloat *a) const
  変換行列を配列に格納した変換行列で除算した値を返す.
- GgMatrix divide (const GgMatrix &m) const
  変換行列を配列に格納した変換行列で除算した値を返す.
- GgMatrix & operator= (const GLfloat *a)
- GgMatrix & operator= (const GgMatrix &m)
- GgMatrix & operator+= (const GLfloat *a)
- GgMatrix & operator+= (const GgMatrix &m)
- GgMatrix & operator-= (const GLfloat *a)

- GgMatrix & operator-= (const GgMatrix &m)
- GgMatrix & operator∗= (const GLfloat ∗a)
- GgMatrix & operator∗= (const GgMatrix &m)
- GgMatrix & operator/= (const GLfloat ∗a)
- GgMatrix & operator/= (const GgMatrix &m)
- GgMatrix operator+ (const GLfloat ∗a) const
- GgMatrix operator+ (const GgMatrix &m) const
- GgMatrix operator- (const GLfloat ∗a) const
- GgMatrix operator- (const GgMatrix &m) const
- GgMatrix operator∗ (const GLfloat ∗a) const
- GgMatrix operator∗ (const GgMatrix &m) const
- GgMatrix operator/ (const GLfloat ∗a) const
- GgMatrix operator/ (const GgMatrix &m) const
- GgMatrix & loadIdentity ()

  単位行列を格納する.
- GgMatrix & loadTranslate (GLfloat x, GLfloat y, GLfloat z, GLfloat w=1.0f)

  平行移動の変換行列を格納する.
- GgMatrix & loadTranslate (const GLfloat ∗t)

  平行移動の変換行列を格納する.
- GgMatrix & loadScale (GLfloat x, GLfloat y, GLfloat z, GLfloat w=1.0f)

  拡大縮小の変換行列を格納する.
- GgMatrix & loadScale (const GLfloat ∗s)

  拡大縮小の変換行列を格納する.
- GgMatrix & loadRotateX (GLfloat a)

  x **軸中心の回転の変換行列を格納する**.
- GgMatrix & loadRotateY (GLfloat a)

  y **軸中心の回転の変換行列を格納する**.
- GgMatrix & loadRotateZ (GLfloat a)

  z **軸中心の回転の変換行列を格納する**.
- GgMatrix & loadRotate (GLfloat x, GLfloat y, GLfloat z, GLfloat a)

  (x, y, z) **方向のベクトルを軸とする回転の変換行列を格納する**.
- GgMatrix & loadRotate (const GLfloat ∗r, GLfloat a)

  r **方向のベクトルを軸とする回転の変換行列を格納する**.
- GgMatrix & loadRotate (const GLfloat ∗r)

  r **方向のベクトルを軸とする回転の変換行列を格納する**.
- GgMatrix & loadLookat (GLfloat ex, GLfloat ey, GLfloat ez, GLfloat tx, GLfloat ty, GLfloat tz, GLfloat ux, GLfloat uy, GLfloat uz)

  ビュー変換行列を格納する.
- GgMatrix & loadLookat (const GLfloat ∗e, const GLfloat ∗t, const GLfloat ∗u)

  ビュー変換行列を格納する.
- GgMatrix & loadOrthogonal (GLfloat left, GLfloat right, GLfloat bottom, GLfloat top, GLfloat zNear, GLfloat zFar)

  直交投影変換行列を格納する.
- GgMatrix & loadFrustum (GLfloat left, GLfloat right, GLfloat bottom, GLfloat top, GLfloat zNear, GLfloat zFar)

  透視透視投影変換行列を格納する.
- GgMatrix & loadPerspective (GLfloat fovy, GLfloat aspect, GLfloat zNear, GLfloat zFar)

  画角を指定して透視投影変換行列を格納する.
- GgMatrix & loadTranspose (const GLfloat ∗a)

  転置行列を格納する.
- GgMatrix & loadTranspose (const GgMatrix &m)

  転置行列を格納する.
- GgMatrix & loadInvert (const GLfloat ∗a)

逆行列を格納する.

- GgMatrix & loadInvert (const GgMatrix &m)

  逆行列を格納する.
- GgMatrix & loadNormal (const GLfloat *a)

  法線変換行列を格納する.
- GgMatrix & loadNormal (const GgMatrix &m)

  法線変換行列を格納する.
- GgMatrix translate (GLfloat x, GLfloat y, GLfloat z, GLfloat w=1.0f) const

  平行移動変換を乗じた結果を返す.
- GgMatrix translate (const GLfloat *t) const

  平行移動変換を乗じた結果を返す.
- GgMatrix scale (GLfloat x, GLfloat y, GLfloat z, GLfloat w=1.0f) const

  拡大縮小変換を乗じた結果を返す.
- GgMatrix scale (const GLfloat *s) const

  拡大縮小変換を乗じた結果を返す.
- GgMatrix rotateX (GLfloat a) const

  x **軸中心の回転変換を乗じた結果を返す**.
- GgMatrix rotateY (GLfloat a) const

  y **軸中心の回転変換を乗じた結果を返す**.
- GgMatrix rotateZ (GLfloat a) const

  z **軸中心の回転変換を乗じた結果を返す**.
- GgMatrix rotate (GLfloat x, GLfloat y, GLfloat z, GLfloat a) const

  (x, y, z) **方向のベクトルを軸とする回転変換を乗じた結果を返す**.
- GgMatrix rotate (const GLfloat *r, GLfloat a) const

  r **方向のベクトルを軸とする回転変換を乗じた結果を返す**.
- GgMatrix rotate (const GLfloat *r) const

  r **方向のベクトルを軸とする回転の変換行列を乗じた結果を返す**.
- GgMatrix lookat (GLfloat ex, GLfloat ey, GLfloat ez, GLfloat tx, GLfloat ty, GLfloat tz, GLfloat ux, GLfloat uy, GLfloat uz) const

  ビュー変換を乗じた結果を返す.
- GgMatrix lookat (const GLfloat *e, const GLfloat *t, const GLfloat *u) const

  ビュー変換を乗じた結果を返す.
- GgMatrix orthogonal (GLfloat left, GLfloat right, GLfloat bottom, GLfloat top, GLfloat zNear, GLfloat zFar) const

  直交投影変換を乗じた結果を返す.
- GgMatrix frustum (GLfloat left, GLfloat right, GLfloat bottom, GLfloat top, GLfloat zNear, GLfloat zFar) const

  透視投影変換を乗じた結果を返す.
- GgMatrix perspective (GLfloat fovy, GLfloat aspect, GLfloat zNear, GLfloat zFar) const

  画角を指定して透視投影変換を乗じた結果を返す.
- GgMatrix transpose () const

  転置行列を返す.
- GgMatrix invert () const

  逆行列を返す.
- GgMatrix normal () const

  法線変換行列を返す.
- void projection (GLfloat *c, const GLfloat *v) const

  ベクトルに対して投影変換を行う.
- const GLfloat * get () const

  変換行列を取り出す.
- void get (GLfloat *a) const

  変換行列を取り出す.

フレンド

- class GgQuaternion

6.5.1 詳解

変換行列.

gg.h の 2805 行目に定義があります。

6.5.2 構築子と解体子

6.5.2.1 gg::GgMatrix::~GgMatrix ( ) [inline]

デストラクタ.

gg.h の 2822 行目に定義があります。

6.5.2.2 gg::GgMatrix::GgMatrix ( ) [inline]

コンストラクタ.

gg.h の 2825 行目に定義があります。

6.5.2.3 gg::GgMatrix::GgMatrix (const GLfloat * a) [inline]

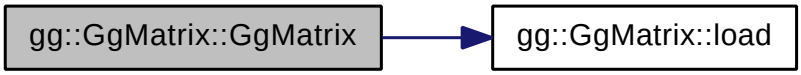
コンストラクタ.

引数

a	GLfloat 型の 16 要素の配列.

gg.h の 2829 行目に定義があります。

呼び出し関係図:



6.5.2.4 gg::GgMatrix::GgMatrix (const GgMatrix & m) [inline]

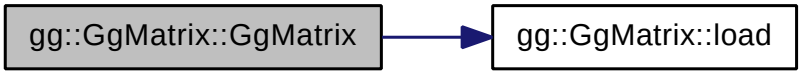
コピーコンストラクタ.

引数

m	GgMatrix 型の変数.

gg.h の 2836 行目に定義があります。

呼び出し関係図:



6.5.3 関数詳解

6.5.3.1 GgMatrix gg::GgMatrix::add (const GLfloat * a) const [inline]

変換行列に配列に格納した変換行列を加算した値を返す.

引数

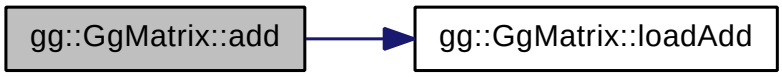
a	GLfloat 型の 16 要素の配列.

戻り値

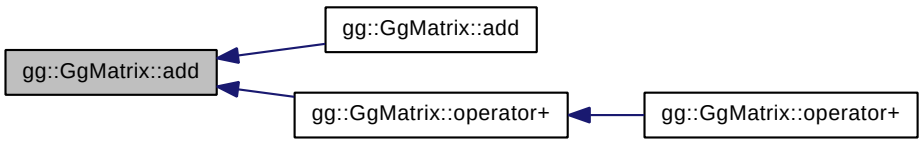
変換行列に a を加えた GgMatrix 型の値.

gg.h の 2927 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:



6.5.3.2 GgMatrix gg::GgMatrix::add (const GgMatrix & m) const [inline]

変換行列に別の変換行列を加算した値を返す.

引数

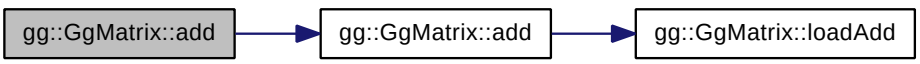
m	GgMatrix 型の変数.

戻り値

変換行列に m を加えた GgMatrix 型の値.

gg.h の 2936 行目に定義があります。

呼び出し関係図:



6.5.3.3 GgMatrix gg::GgMatrix::divide (const GLfloat * a) const [inline]

変換行列を配列に格納した変換行列で除算した値を返す.

引数

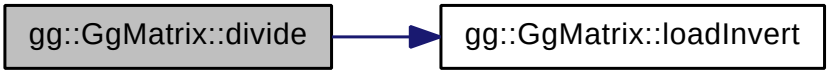
a	GLfloat 型の 16 要素の配列.

戻り値

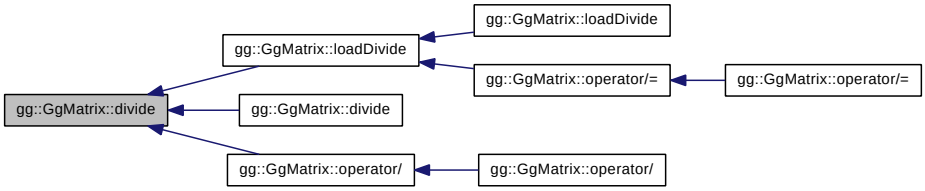
変換行列を a で割った GgMatrix 型の値.

gg.h の 2979 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:



6.5.3.4 GgMatrix gg::GgMatrix::divide (const GgMatrix & *m*) const [inline]

変換行列を配列に格納した変換行列で除算した値を返す.

引数

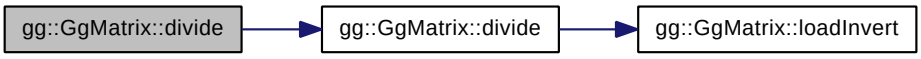
m	GgMatrix 型の変数.

戻り値

変換行列を m で割った GgMatrix 型の値.

gg.h の 2990 行目に定義があります。

呼び出し関係図:



6.5.3.5 GgMatrix gg::GgMatrix::frustum (GLfloat *left*, GLfloat *right*, GLfloat *bottom*, GLfloat *top*, GLfloat *zNear*, GLfloat *zFar*) const [inline]

透視投影変換を乗じた結果を返す.

引数

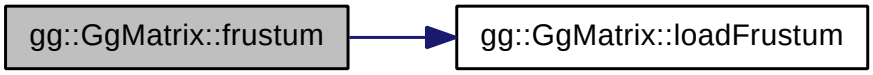
left	ウィンドウの左端の位置.
right	ウィンドウの右端の位置.
bottom	ウィンドウの下端の位置.
top	ウィンドウの上端の位置.
zNear	視点から前方面までの位置.
zFar	視点から後方面までの位置.

戻り値

透視投影変換行列を乗じた変換行列.

gg.h の 3390 行目に定義があります。

呼び出し関係図:



6.5.3.6 const GLfloat* gg::GgMatrix::get ( ) const [inline]

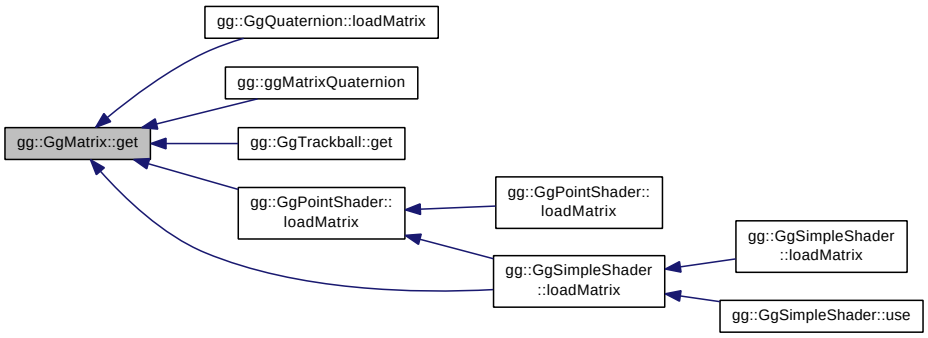
変換行列を取り出す.

戻り値

変換行列を格納した GLfloat 型の 16 要素の配列.

gg.h の 3445 行目に定義があります。

被呼び出し関係図:



6.5.3.7 void gg::GgMatrix::get (GLfloat * *a*) const [inline]

変換行列を取り出す.

引数

a	変換行列を格納する GLfloat 型の 16 要素の配列.

gg.h の 3452 行目に定義があります。

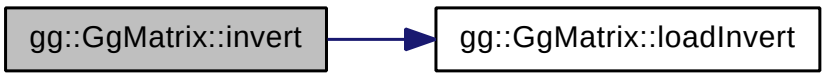
6.5.3.8 GgMatrix gg::GgMatrix::invert ( ) const [inline]

逆行列を返す.

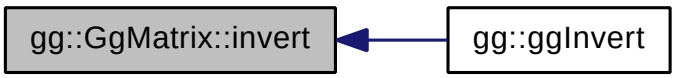
戻り値

逆行列.

gg.h の 3421 行目に定義があります。
呼び出し関係図:



被呼び出し関係図:



6.5.3.9 GgMatrix& gg::GgMatrix::load (const GLfloat * *a*) [inline]

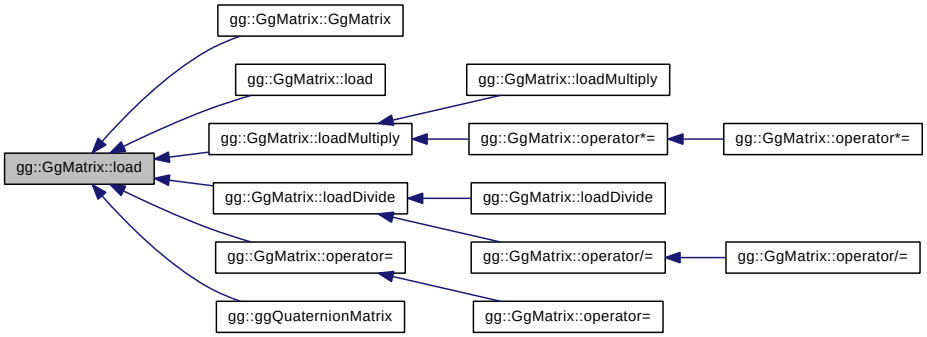
配列変数の値を格納する.
引数

a	GLfloat 型の 16 要素の配列.

戻り値

a を代入した GgMatrix 型の値.

gg.h の 2844 行目に定義があります。
被呼び出し関係図:



6.5.3.10 GgMatrix& gg::GgMatrix::load (const GgMatrix & *m*) [inline]

別の変換行列の値を格納する.

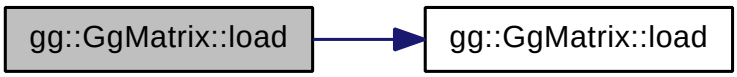
引数

m	GgMatrix 型の変数.

戻り値

m を代入した GgMatrix 型の値.

gg.h の 2853 行目に定義があります。
呼び出し関係図:



6.5.3.11 GgMatrix& gg::GgMatrix::loadAdd (const GLfloat * *a*) [inline]

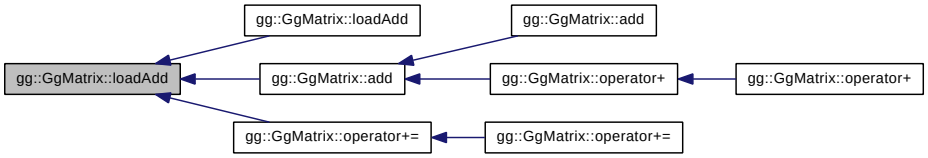
変換行列に配列に格納した変換行列を加算した結果を格納する.
引数

a	GLfloat 型の 16 要素の配列.

戻り値

変換行列に a を加えた GgMatrix 型の値.

gg.h の 2861 行目に定義があります。
被呼び出し関係図:



6.5.3.12 GgMatrix& gg::GgMatrix::loadAdd (const GgMatrix & *m*) [inline]

変換行列に別の変換行列を加算した結果を格納する.
引数

m	GgMatrix 型の変数.

戻り値

変換行列に m を加えた GgMatrix 型の値.

gg.h の 2870 行目に定義があります。

呼び出し関係図:

6.5.3.13 GgMatrix& gg::GgMatrix::loadDivide (const GLfloat * *a*) [inline]

変換行列を配列に格納した変換行列で除算した結果を格納する.

引数

a	GLfloat 型の 16 要素の配列.

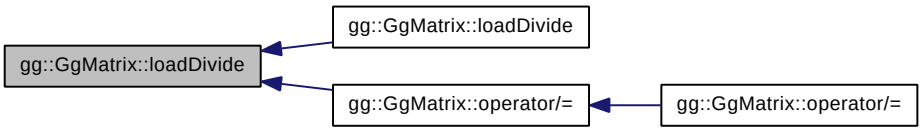
戻り値

変換行列に a を乗じた GgMatrix 型の値.

gg.h の 2911 行目に定義があります。

呼び出し関係図:

被呼び出し関係図:



6.5.3.14 GgMatrix& gg::GgMatrix::loadDivide (const GgMatrix & *m*) [inline]

変換行列を別の変換行列で除算した結果を格納する.

引数

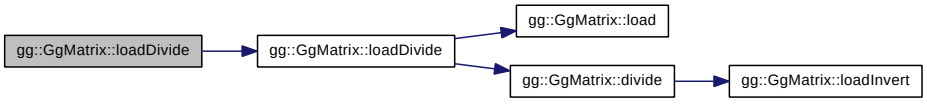
m	GgMatrix 型の変数.

戻り値

変換行列に m を乗じた GgMatrix 型の値.

gg.h の 2919 行目に定義があります。

呼び出し関係図:



6.5.3.15 gg::GgMatrix & gg::GgMatrix::loadFrustum (GLfloat *left*, GLfloat *right*, GLfloat *bottom*, GLfloat *top*, GLfloat *zNear*, GLfloat *zFar*)

透視透視投影変換行列を格納する.

引数

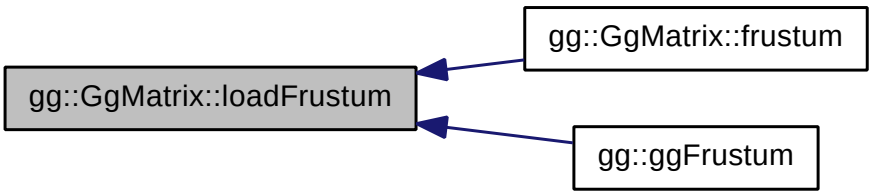
left	ウィンドウの左端の位置.
right	ウィンドウの右端の位置.
bottom	ウィンドウの下端の位置.
top	ウィンドウの上端の位置.
zNear	視点から前方面までの位置.
zFar	視点から後方面までの位置.

戻り値

設定した透視投影変換行列.

gg.cpp の 7137 行目に定義があります。

被呼び出し関係図:



6.5.3.16 gg::GgMatrix & gg::GgMatrix::loadIdentity ( )

単位行列を格納する.

gg.cpp の 6795 行目に定義があります。

被呼び出し関係図:

6.5.3.17 gg::GgMatrix & gg::GgMatrix::loadInvert (const GLfloat * *a*)

逆行列を格納する.

引数

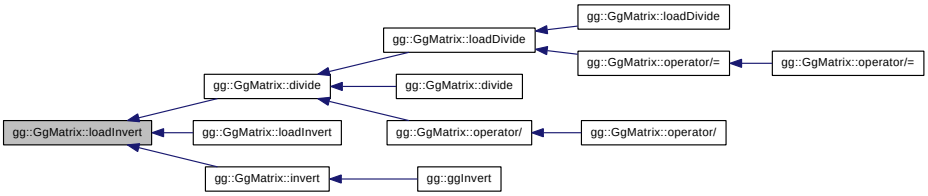
a	GLfloat 型の 16 要素の変換行列.

戻り値

設定した a の逆行列.

gg.cpp の 6952 行目に定義があります。

被呼び出し関係図:



6.5.3.18 GgMatrix& gg::GgMatrix::loadInvert (const GgMatrix & *m*) [inline]

逆行列を格納する.

引数

m	GgMatrix 型の変換行列.

戻り値

設定した m の逆行列.

gg.h の 3223 行目に定義があります。

呼び出し関係図:

6.5.3.19 gg::GgMatrix & gg::GgMatrix::loadLookat (GLfloat *ex*, GLfloat *ey*, GLfloat *ez*, GLfloat *tx*, GLfloat *ty*, GLfloat *tz*, GLfloat *ux*, GLfloat *uy*, GLfloat *uz*)

ビュー変換行列を格納する.

引数

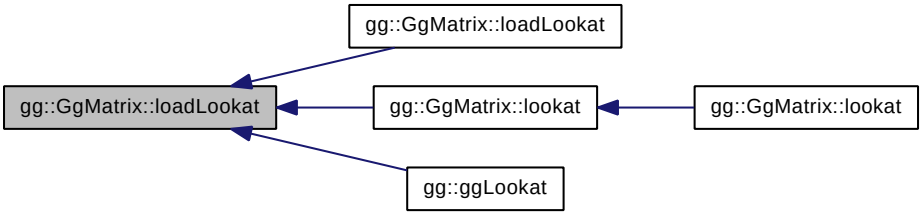
ex	視点の位置の x 座標値.
ey	視点の位置の y 座標値.
ez	視点の位置の z 座標値.
tx	目標点の位置の x 座標値.
ty	目標点の位置の y 座標値.
tz	目標点の位置の z 座標値.
ux	上方向のベクトルの x 成分.
uy	上方向のベクトルの y 成分.
uz	上方向のベクトルの z 成分.

戻り値

設定したビュー変換行列.

gg.cpp の 7054 行目に定義があります。

被呼び出し関係図:



6.5.3.20 GgMatrix& gg::GgMatrix::loadLookat (const GLfloat * e, const GLfloat * t, const GLfloat * u) [inline]

ビュー変換行列を格納する.

引数

e	視点の位置の配列変数.
t	目標点の位置の配列変数.
u	上方向のベクトルのの配列変数.

戻り値

設定したビュー変換行列.

gg.h の 3164 行目に定義があります。

呼び出し関係図:

6.5.3.21 GgMatrix& gg::GgMatrix::loadMultiply (const GLfloat * a) [inline]

変換行列に配列に格納した変換行列を乗算した結果を格納する.

引数

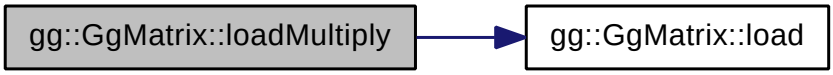
a	GLfloat 型の 16 要素の配列.

戻り値

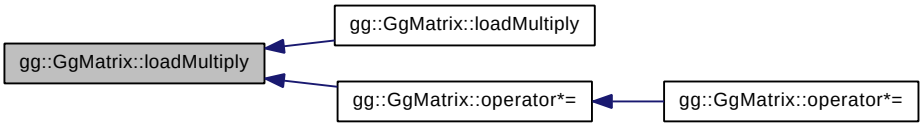
変換行列に a を掛けた GgMatrix 型の値.

gg.h の 2895 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:



6.5.3.22 GgMatrix& gg::GgMatrix::loadMultiply (const GgMatrix & *m*) [inline]

変換行列に別の変換行列を乗算した結果を格納する.

引数

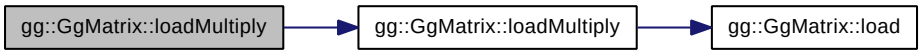
m	GgMatrix 型の変数.

戻り値

変換行列に m を掛けた GgMatrix 型の値.

gg.h の 2903 行目に定義があります。

呼び出し関係図:



6.5.3.23 gg::GgMatrix & gg::GgMatrix::loadNormal (const GLfloat * *a*)

法線変換行列を格納する.

引数

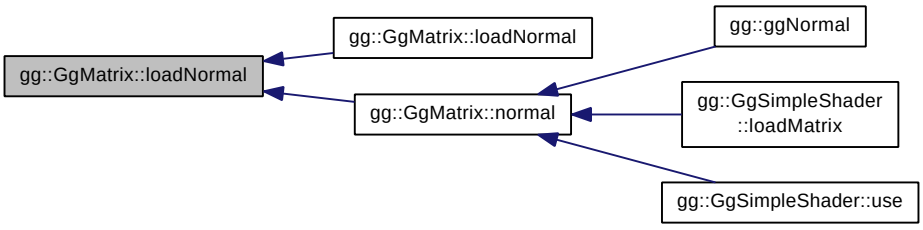
a	GLfloat 型の 16 要素の変換行列.

戻り値

設定した m の法線変換行列.

gg.cpp の 7034 行目に定義があります。

被呼び出し関係図:



6.5.3.24 GgMatrix& gg::GgMatrix::loadNormal (const GgMatrix & *m*) [inline]

法線変換行列を格納する.

引数

GgMatrix	型の変換行列.

戻り値

設定した m の法線変換行列.

gg.h の 3236 行目に定義があります。

呼び出し関係図:

6.5.3.25 gg::GgMatrix & gg::GgMatrix::loadOrthogonal (GLfloat *left*, GLfloat *right*, GLfloat *bottom*, GLfloat *top*, GLfloat *zNear*, GLfloat *zFar*)

直交投影変換行列を格納する.

引数

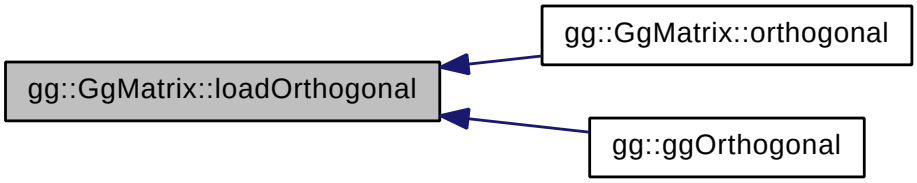
left	ウィンドウの左端の位置.
right	ウィンドウの右端の位置.
bottom	ウィンドウの下端の位置.
top	ウィンドウの上端の位置.
zNear	視点から前方面までの位置.
zFar	視点から後方面までの位置.

戻り値

設定した直交投影変換行列.

gg.cpp の 7110 行目に定義があります。

被呼び出し関係図:



6.5.3.26 gg::GgMatrix & gg::GgMatrix::loadPerspective (GLfloat *fovy*, GLfloat *aspect*, GLfloat *zNear*, GLfloat *zFar*)

画角を指定して透視投影変換行列を格納する.

引数

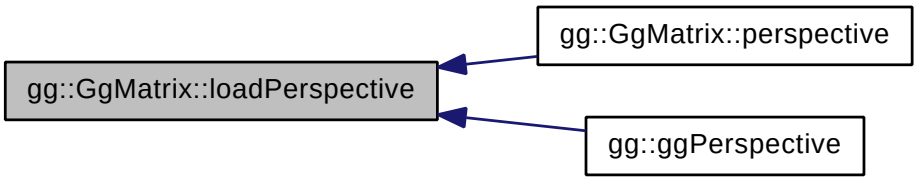
fovy	y 方向の画角.
aspect	縦横比.
zNear	視点から前方面までの位置.
zFar	視点から後方面までの位置.

戻り値

設定した透視投影変換行列.

gg.cpp の 7164 行目に定義があります。

被呼び出し関係図:



6.5.3.27 gg::GgMatrix & gg::GgMatrix::loadRotate (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *a*)

(x, y, z) 方向のベクトルを軸とする回転の変換行列を格納する.

引数

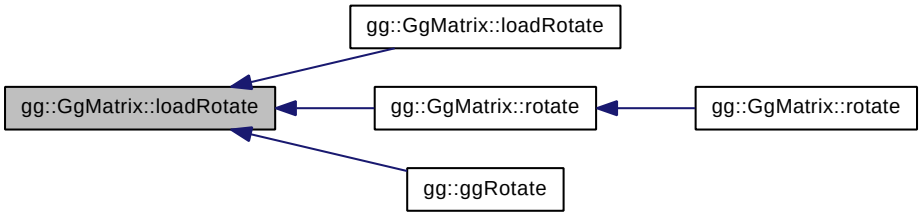
x	回転軸の x 成分.
y	回転軸の y 成分.
z	回転軸の z 成分.
a	回転角.

戻り値

設定した変換行列.

gg.cpp の 6888 行目に定義があります。

被呼び出し関係図:



6.5.3.28 GgMatrix& gg::GgMatrix::loadRotate (const GLfloat * *r*, GLfloat *a*) [inline]

r 方向のベクトルを軸とする回転の変換行列を格納する.

引数

r	回転軸の方向ベクトル (x, y, z).
a	回転角.

戻り値

設定した変換行列.

gg.h の 3131 行目に定義があります。

呼び出し関係図:

6.5.3.29 GgMatrix& gg::GgMatrix::loadRotate (const GLfloat * *r*) [inline]

r 方向のベクトルを軸とする回転の変換行列を格納する.

引数

r	回転軸の方向ベクトルと回転角 (x, y, z, a).

戻り値

設定した変換行列.

gg.h の 3139 行目に定義があります。

呼び出し関係図:

6.5.3.30 gg::GgMatrix & gg::GgMatrix::loadRotateX (GLfloat *a*)

x 軸中心の回転の変換行列を格納する.

引数

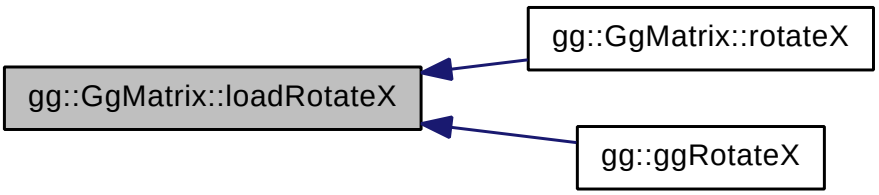
a	回転角.

戻り値

設定した変換行列.

gg.cpp の 6840 行目に定義があります。

被呼び出し関係図:



6.5.3.31 gg::GgMatrix & gg::GgMatrix::loadRotateY (GLfloat *a*)

y 軸中心の回転の変換行列を格納する.

引数

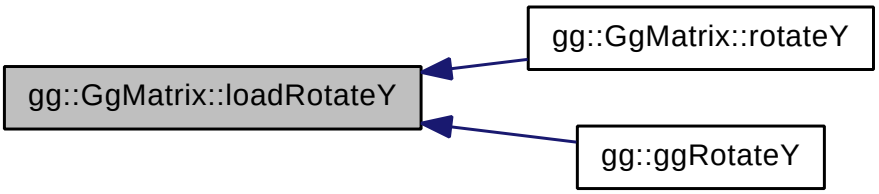
a	回転角.

戻り値

設定した変換行列.

gg.cpp の 6856 行目に定義があります。

被呼び出し関係図:



6.5.3.32 gg::GgMatrix & gg::GgMatrix::loadRotateZ (GLfloat *a*)

z 軸中心の回転の変換行列を格納する.

引数

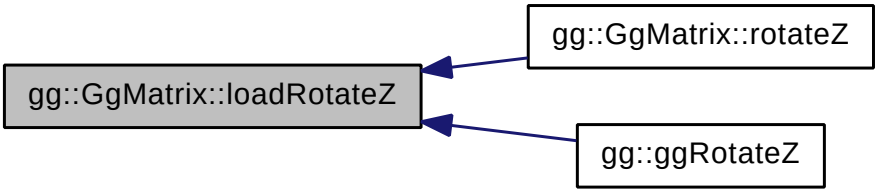
a	回転角.

戻り値

設定した変換行列.

gg.cpp の 6872 行目に定義があります。

被呼び出し関係図:



6.5.3.33 gg::GgMatrix & gg::GgMatrix::loadScale (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *w* = `1.0f`)

拡大縮小の変換行列を格納する.

引数

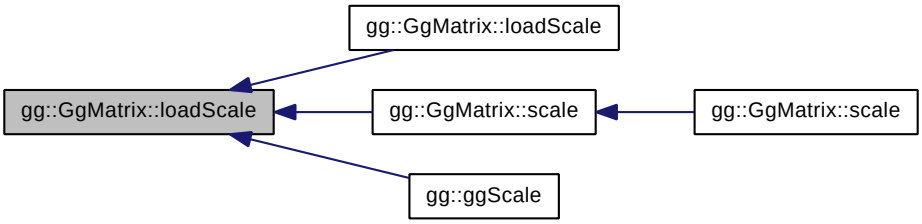
x	x 方向の拡大率.
y	y 方向の拡大率.
z	z 方向の拡大率.
w	w 拡大率のスケールファクタ (= 1.0f).

戻り値

設定した変換行列.

gg.cpp の 6824 行目に定義があります。

被呼び出し関係図:



6.5.3.34 GgMatrix& gg::GgMatrix::loadScale (const GLfloat * *s*) `[inline]`

拡大縮小の変換行列を格納する.

引数

s	拡大率の GLfloat 型の配列 (x, y, z).

戻り値

設定した変換行列.

gg.h の 3099 行目に定義があります。

呼び出し関係図:

6.5.3.35 GgMatrix& gg::GgMatrix::loadSubtract (const GLfloat * a) [inline]

変換行列から配列に格納した変換行列を減算した結果を格納する.

引数

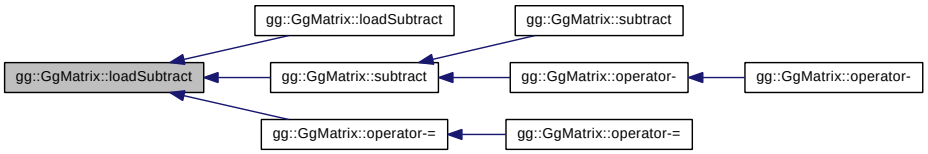
a	GLfloat 型の 16 要素の配列.

戻り値

変換行列に a を引いた GgMatrix 型の値.

gg.h の 2878 行目に定義があります。

被呼び出し関係図:



6.5.3.36 GgMatrix& gg::GgMatrix::loadSubtract (const GgMatrix & m) [inline]

変換行列から別の変換行列を減算した結果を格納する.

引数

m	GgMatrix 型の変数.

戻り値

変換行列に m を引いた GgMatrix 型の値.

gg.h の 2887 行目に定義があります。

呼び出し関係図:

6.5.3.37 gg::GgMatrix & gg::GgMatrix::loadTranslate (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *w* = 1.0f)

平行移動の変換行列を格納する.

引数

x	x 方向の移動量.
y	y 方向の移動量.
z	z 方向の移動量.

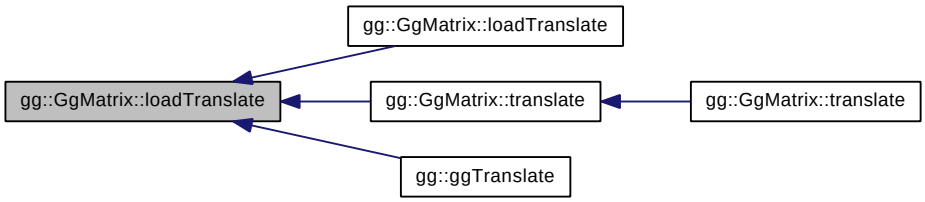
w	w 移動量のスケールファクタ (= 1.0f).

戻り値

設定した変換行列.

gg.cpp の 6808 行目に定義があります。

被呼び出し関係図:



6.5.3.38 GgMatrix& gg::GgMatrix::loadTranslate (const GLfloat * *t*) [inline]

平行移動の変換行列を格納する.

引数

t	移動量の GLfloat 型の配列 (x, y, z).

戻り値

設定した変換行列.

gg.h の 3083 行目に定義があります。

呼び出し関係図:

6.5.3.39 gg::GgMatrix & gg::GgMatrix::loadTranspose (const GLfloat * *a*)

転置行列を格納する.

引数

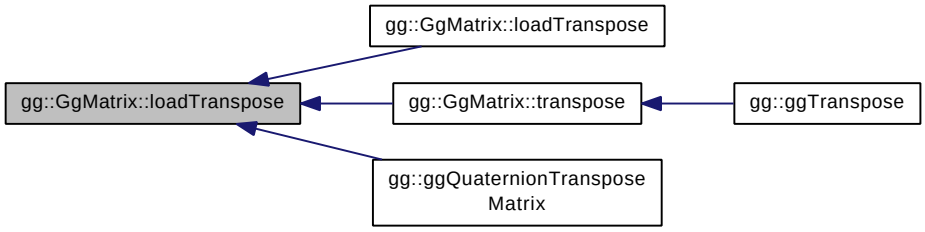
a	GLfloat 型の 16 要素の変換行列.

戻り値

設定した a の転置行列.

gg.cpp の 6927 行目に定義があります。

被呼び出し関係図:



6.5.3.40 GgMatrix& gg::GgMatrix::loadTranspose (const GgMatrix & *m*) [inline]

転置行列を格納する.

引数

m	GgMatrix 型の変換行列.

戻り値

設定した m の転置行列.

gg.h の 3210 行目に定義があります。

呼び出し関係図:

6.5.3.41 GgMatrix gg::GgMatrix::lookat (GLfloat *ex*, GLfloat *ey*, GLfloat *ez*, GLfloat *tx*, GLfloat *ty*, GLfloat *tz*, GLfloat *ux*, GLfloat *uy*, GLfloat *uz*) const [inline]

ビュー変換を乗じた結果を返す.

引数

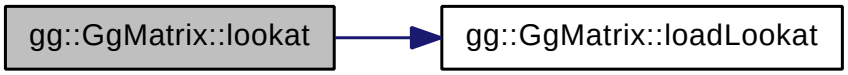
ex	視点の位置の x 座標値.
ey	視点の位置の y 座標値.
ez	視点の位置の z 座標値.
tx	目標点の位置の x 座標値.
ty	目標点の位置の y 座標値.
tz	目標点の位置の z 座標値.
ux	上方向のベクトルの x 成分.
uy	上方向のベクトルの y 成分.
uz	上方向のベクトルの z 成分.

戻り値

ビュー変換行列を乗じた変換行列.

gg.h の 3348 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:

6.5.3.42 GgMatrix gg::GgMatrix::lookat (const GLfloat * e, const GLfloat * t, const GLfloat * u) const [inline]

ビュー変換を乗じた結果を返す.

引数

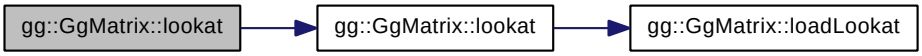
e	視点の位置の配列変数.
t	目標点の位置の配列変数.
u	上方向のベクトルのの配列変数.

戻り値

ビュー変換行列を乗じた変換行列.

gg.h の 3361 行目に定義があります。

呼び出し関係図:



6.5.3.43 GgMatrix gg::GgMatrix::multiply (const GLfloat * a) const [inline]

変換行列に配列に格納した変換行列を乗算した値を返す.

引数

a	GLfloat 型の 16 要素の配列.

戻り値

変換行列に a を掛けた GgMatrix 型の値.

gg.h の 2961 行目に定義があります。

6.5.3.44 GgMatrix gg::GgMatrix::multiply (const GgMatrix & m) const [inline]

変換行列に別の変換行列を乗算した値を返す.

引数

m	GgMatrix 型の変数.

戻り値

変換行列に m を掛けた GgMatrix 型の値.

gg.h の 2971 行目に定義があります。

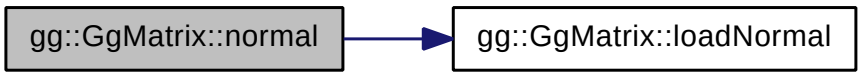
6.5.3.45 GgMatrix gg::GgMatrix::normal ( ) const [inline]

法線変換行列を返す.

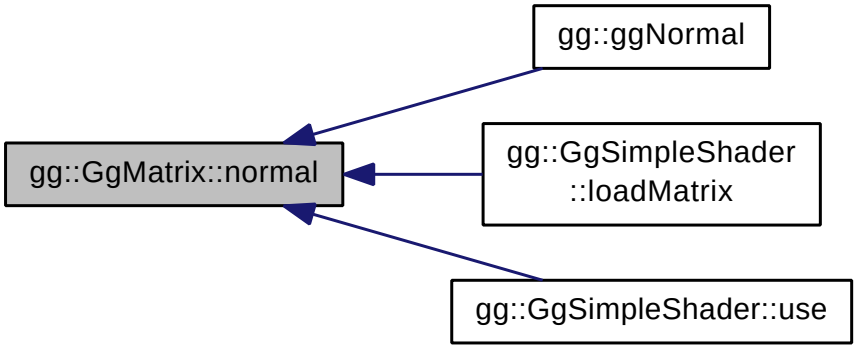
戻り値

法線変換行列.

gg.h の 3429 行目に定義があります。
呼び出し関係図:



被呼び出し関係図:



6.5.3.46 GgMatrix gg::GgMatrix::operator∗ (const GLfloat ∗ *a*) const [inline]

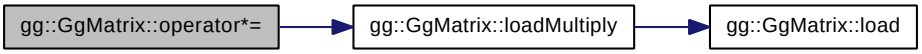
gg.h の 3052 行目に定義があります。
被呼び出し関係図:

6.5.3.47 GgMatrix gg::GgMatrix::operator∗ (const GgMatrix & *m*) const [inline]

gg.h の 3056 行目に定義があります。
呼び出し関係図:

6.5.3.48 GgMatrix& gg::GgMatrix::operator∗= (const GLfloat ∗ *a*) [inline]

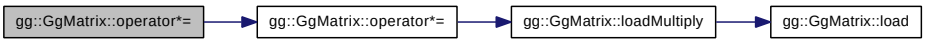
gg.h の 3020 行目に定義があります。
呼び出し関係図:



被呼び出し関係図:

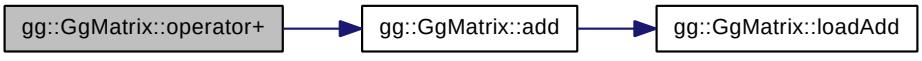
6.5.3.49 GgMatrix& gg::GgMatrix::operator∗= (const GgMatrix & *m*) [inline]

gg.h の 3024 行目に定義があります。
呼び出し関係図:



6.5.3.50 GgMatrix gg::GgMatrix::operator+ (const GLfloat ∗ *a*) const [inline]

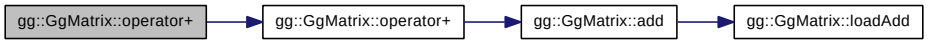
gg.h の 3036 行目に定義があります。
呼び出し関係図:



被呼び出し関係図:

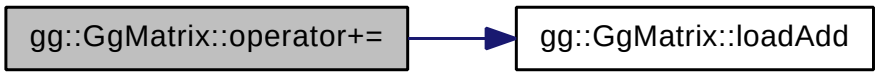
6.5.3.51 GgMatrix gg::GgMatrix::operator+ (const GgMatrix & *m*) const [inline]

gg.h の 3040 行目に定義があります。
呼び出し関係図:



6.5.3.52 GgMatrix& gg::GgMatrix::operator+= (const GLfloat * *a*) [inline]

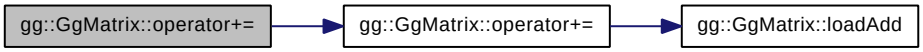
gg.h の 3004 行目に定義があります。
呼び出し関係図:



被呼び出し関係図:

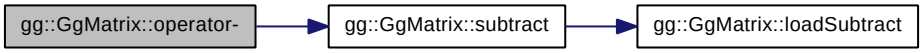
6.5.3.53 GgMatrix& gg::GgMatrix::operator+= (const GgMatrix & *m*) [inline]

gg.h の 3008 行目に定義があります。
呼び出し関係図:



6.5.3.54 GgMatrix gg::GgMatrix::operator- (const GLfloat * *a*) const [inline]

gg.h の 3044 行目に定義があります。
呼び出し関係図:



被呼び出し関係図:

6.5.3.55 GgMatrix gg::GgMatrix::operator- (const GgMatrix & *m*) const [inline]

gg.h の 3048 行目に定義があります。
呼び出し関係図:

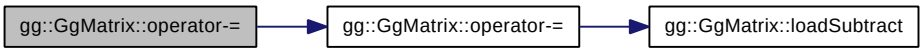
6.5.3.56 GgMatrix& gg::GgMatrix::operator-= (const GLfloat * *a*) [inline]

gg.h の 3012 行目に定義があります。
呼び出し関係図:

被呼び出し関係図:

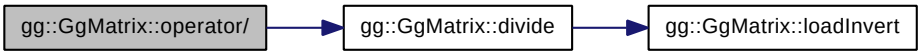
6.5.3.57 GgMatrix& gg::GgMatrix::operator-= (const GgMatrix & *m*) [inline]

gg.h の 3016 行目に定義があります。
呼び出し関係図:



6.5.3.58 GgMatrix gg::GgMatrix::operator/ (const GLfloat * *a*) const [inline]

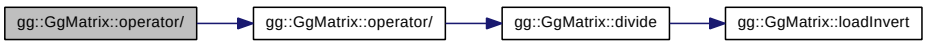
gg.h の 3060 行目に定義があります。
呼び出し関係図:



被呼び出し関係図:

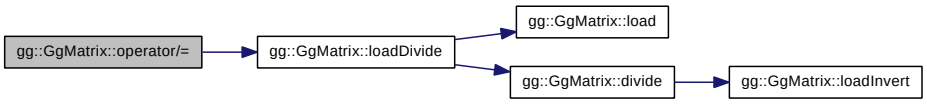
6.5.3.59 GgMatrix gg::GgMatrix::operator/ (const GgMatrix & *m*) const [inline]

gg.h の 3064 行目に定義があります。
呼び出し関係図:



6.5.3.60 GgMatrix& gg::GgMatrix::operator/= (const GLfloat * *a*) [inline]

gg.h の 3028 行目に定義があります。
呼び出し関係図:



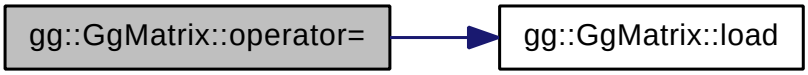
被呼び出し関係図:

6.5.3.61 GgMatrix& gg::GgMatrix::operator/= (const GgMatrix & *m*) [inline]

gg.h の 3032 行目に定義があります。
呼び出し関係図:

6.5.3.62 GgMatrix& gg::GgMatrix::operator= (const GLfloat * *a*) [inline]

gg.h の 2996 行目に定義があります。
呼び出し関係図:

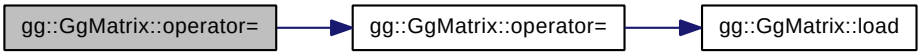


被呼び出し関係図:

6.5.3.63 GgMatrix& gg::GgMatrix::operator= (const GgMatrix & *m*) [inline]

gg.h の 3000 行目に定義があります。

呼び出し関係図:



6.5.3.64 GgMatrix gg::GgMatrix::orthogonal (GLfloat *left*, GLfloat *right*, GLfloat *bottom*, GLfloat *top*, GLfloat *zNear*, GLfloat *zFar*) const [inline]

直交投影変換を乗じた結果を返す.

引数

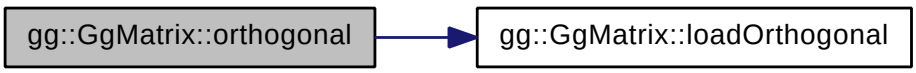
left	ウィンドウの左端の位置.
right	ウィンドウの右端の位置.
bottom	ウィンドウの下端の位置.
top	ウィンドウの上端の位置.
zNear	視点から前方面までの位置.
zFar	視点から後方面までの位置.

戻り値

直交投影変換行列を乗じた変換行列.

gg.h の 3374 行目に定義があります。

呼び出し関係図:



6.5.3.65 GgMatrix gg::GgMatrix::perspective (GLfloat *fovy*, GLfloat *aspect*, GLfloat *zNear*, GLfloat *zFar*) const [inline]

画角を指定して透視投影変換を乗じた結果を返す.

引数

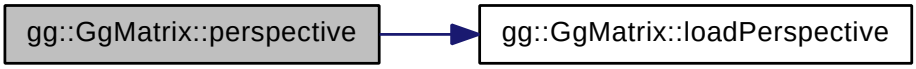
fovy	y 方向の画角.
aspect	縦横比.
zNear	視点から前方面までの位置.
zFar	視点から後方面までの位置.

戻り値

透視投影変換行列を乗じた変換行列.

gg.h の 3404 行目に定義があります。

呼び出し関係図:



6.5.3.66 void gg::GgMatrix::projection (GLfloat * c, const GLfloat * v) const [inline]

ベクトルに対して投影変換を行う.

引数

c	変換結果を格納する GLfloat 型の 4 要素の配列.
v	元のベクトルの GLfloat 型の 4 要素の配列.

gg.h の 3438 行目に定義があります。

6.5.3.67 GgMatrix gg::GgMatrix::rotate (GLfloat x, GLfloat y, GLfloat z, GLfloat a) const [inline]

(x, y, z) 方向のベクトルを軸とする回転変換を乗じた結果を返す.

引数

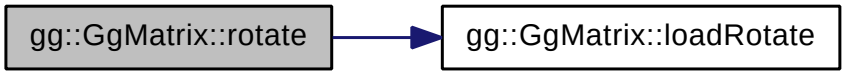
x	回転軸の x 成分.
y	回転軸の y 成分.
z	回転軸の z 成分.
a	回転角.

戻り値

(x, y, z) を軸にさらに a 回転した変換行列.

gg.h の 3314 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:

6.5.3.68 GgMatrix gg::GgMatrix::rotate (const GLfloat * r, GLfloat a) const [inline]

r 方向のベクトルを軸とする回転変換を乗じた結果を返す.

引数

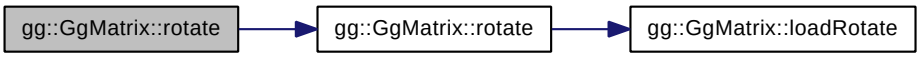
r	回転軸の方向ベクトルと回転角 (x, y, z).
a	回転角.

戻り値

(r[0], r[1], r[2]) を軸にさらに a 回転した変換行列.

gg.h の 3324 行目に定義があります。

呼び出し関係図:



6.5.3.69 GgMatrix gg::GgMatrix::rotate (const GLfloat * *r*) const [inline]

r 方向のベクトルを軸とする回転の変換行列を乗じた結果を返す.

引数

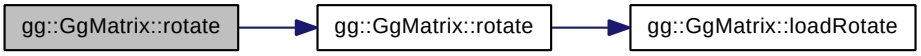
r	回転軸の方向ベクトルと回転角 (x, y, z, a).

戻り値

(r[0], r[1], r[2]) を軸にさらに r[3] 回転した変換行列.

gg.h の 3332 行目に定義があります。

呼び出し関係図:



6.5.3.70 GgMatrix gg::GgMatrix::rotateX (GLfloat *a*) const [inline]

x 軸中心の回転変換を乗じた結果を返す.

引数

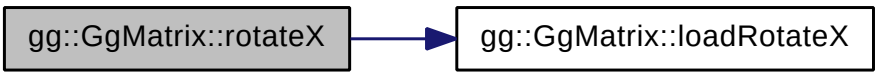
a	回転角.

戻り値

x 軸中心にさらに a 回転した変換行列.

gg.h の 3284 行目に定義があります。

呼び出し関係図:



6.5.3.71 GgMatrix gg::GgMatrix::rotateY (GLfloat *a*) const [inline]

y 軸中心の回転変換を乗じた結果を返す.

引数

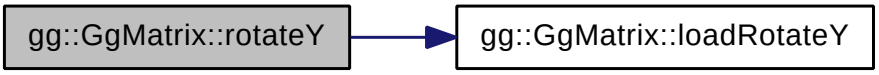
a	回転角.

戻り値

y 軸中心にさらに a 回転した変換行列.

gg.h の 3293 行目に定義があります。

呼び出し関係図:



6.5.3.72 GgMatrix gg::GgMatrix::rotateZ (GLfloat *a*) const [inline]

z 軸中心の回転変換を乗じた結果を返す.

引数

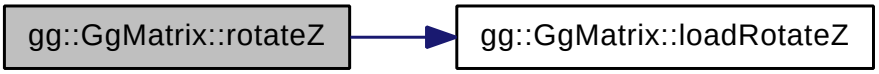
a	回転角.

戻り値

z 軸中心にさらに a 回転した変換行列.

gg.h の 3302 行目に定義があります。

呼び出し関係図:



6.5.3.73 GgMatrix gg::GgMatrix::scale (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *w* = 1.0f) const [inline]

拡大縮小変換を乗じた結果を返す.

引数

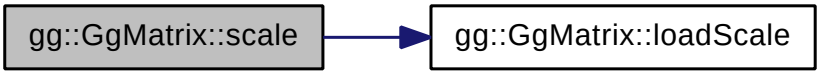
x	x 方向の拡大率.
y	y 方向の拡大率.
z	z 方向の拡大率.
w	w 移動量のスケールファクタ (= 1.0f).

戻り値

拡大縮小した結果の変換行列.

gg.h の 3267 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:

6.5.3.74 GgMatrix gg::GgMatrix::scale (const GLfloat * *s*) const [inline]

拡大縮小変換を乗じた結果を返す.

引数

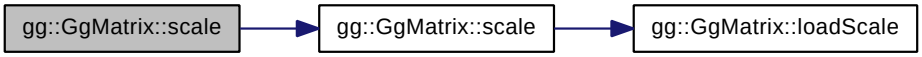
s	拡大率の GLfloat 型の 4 要素の配列 (x, y, z, w).

戻り値

拡大縮小した結果の変換行列.

gg.h の 3276 行目に定義があります。

呼び出し関係図:



6.5.3.75 GgMatrix gg::GgMatrix::subtract (const GLfloat * *a*) const [inline]

変換行列から配列に格納した変換行列を減算した値を返す.

引数

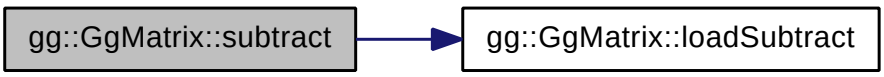
a	GLfloat 型の 16 要素の配列.

戻り値

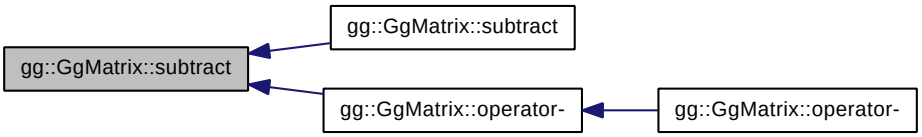
変換行列に a を引いた GgMatrix 型の値.

gg.h の 2944 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:



6.5.3.76 GgMatrix gg::GgMatrix::subtract (const GgMatrix & *m*) const [inline]

変換行列から別の変換行列を減算した値を返す.

引数

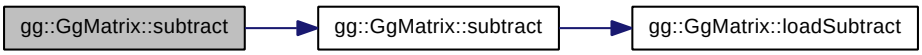
m	GgMatrix 型の変数.

戻り値

変換行列に m を引いた GgMatrix 型の値.

gg.h の 2953 行目に定義があります。

呼び出し関係図:



6.5.3.77 GgMatrix gg::GgMatrix::translate (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *w* = `1.0f`) const [inline]

平行移動変換を乗じた結果を返す.

引数

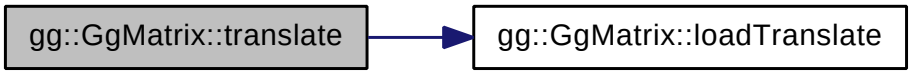
x	x 方向の移動量.
y	y 方向の移動量.
z	z 方向の移動量.
w	w 移動量のスケールファクタ (= 1.0f).

戻り値

平行移動した結果の変換行列.

gg.h の 3247 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:

6.5.3.78 GgMatrix gg::GgMatrix::translate (const GLfloat * *t*) const [inline]

平行移動変換を乗じた結果を返す.

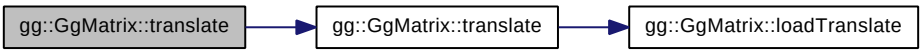
引数

t	移動量の GLfloat 型の 4 要素の配列 (x, y, z, w).

戻り値

平行移動した結果の変換行列.

gg.h の 3256 行目に定義があります。
呼び出し関係図:



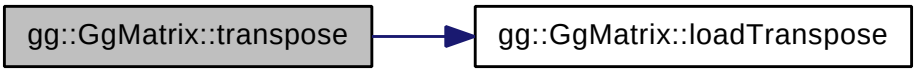
6.5.3.79 GgMatrix gg::GgMatrix::transpose ( ) const [inline]

転置行列を返す.

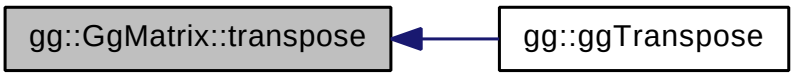
戻り値

転置行列.

gg.h の 3413 行目に定義があります。
呼び出し関係図:



被呼び出し関係図:



6.5.4 フレンドと関連関数の詳解

6.5.4.1 friend class GgQuaternion [friend]

gg.h の 2817 行目に定義があります。
このクラス詳解は次のファイルから抽出されました:

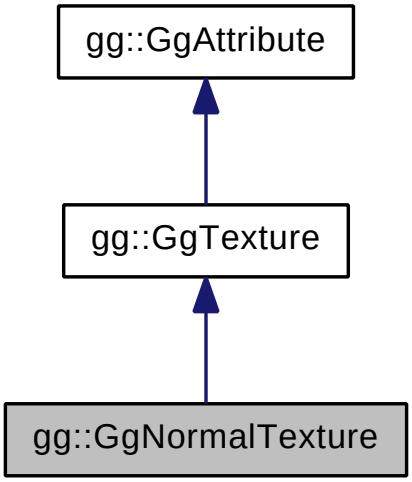
- gg.h
- gg.cpp

6.6 gg::GgNormalTexture クラス

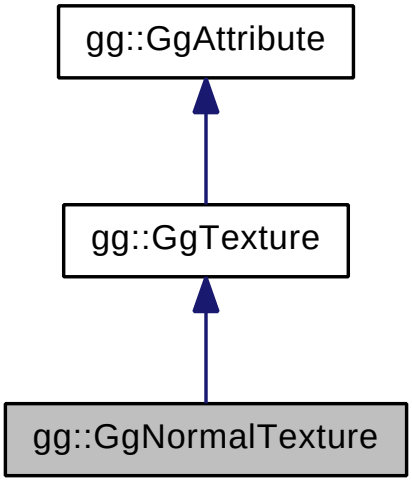
法線マップ.

`#include <gg.h>`

gg::GgNormalTexture の継承関係図



gg::GgNormalTexture 連携図



公開メンバ関数

- virtual ~GgNormalTexture ()
  デストラクタ.
- GgNormalTexture ()
  コンストラクタ.
- GgNormalTexture (const char *name, float nz=1.0f)
  コンストラクタ.
- GgNormalTexture (const GgNormalTexture &o)
  コピーコンストラクタ.
- GgNormalTexture & operator= (const GgNormalTexture &o)

その他の継承メンバ

6.6.1 詳解

法線マップ.
高さマップ（グレイスケール画像）を読み込んで法線マップを作成する.
gg.h の 4874 行目に定義があります。

6.6.2 構築子と解体子

6.6.2.1 virtual gg::GgNormalTexture::~GgNormalTexture ( ) [inline], [virtual]

デストラクタ.
gg.h の 4880 行目に定義があります。

6.6.2.2 gg::GgNormalTexture::GgNormalTexture ( ) [inline]

コンストラクタ.
gg.h の 4883 行目に定義があります。

6.6.2.3 gg::GgNormalTexture::GgNormalTexture (const char * *name*, float *nz* = 1.0f) [inline]

コンストラクタ.
引数

name	画像ファイル名 (1 チャネルの TGA 画像).
nz	法線マップの z 成分の値.

gg.h の 4888 行目に定義があります。

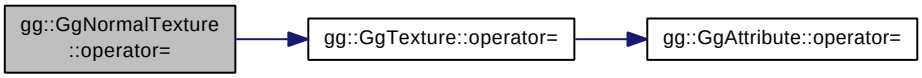
6.6.2.4 gg::GgNormalTexture::GgNormalTexture (const GgNormalTexture & *o*) [inline]

コピーコンストラクタ.
gg.h の 4892 行目に定義があります。

6.6.3 関数詳解

6.6.3.1 GgNormalTexture& gg::GgNormalTexture::operator= (const GgNormalTexture & *o*) [inline]

gg.h の 4896 行目に定義があります。
呼び出し関係図:



このクラス詳解は次のファイルから抽出されました:

- gg.h

6.7 gg::GgObj クラス

Wavefront OBJ 形式のファイル.

`#include <gg.h>`

公開メンバ関数

- virtual ~GgObj ()
  デストラクタ.
- GgObj (const char *name, bool normalize=false)
  コンストラクタ.
- const GgTriangles * get () const
  形状データの取り出し.
- virtual void draw (const GgSimpleShader *shader=NULL, GLint first=0, GLsizei count=0) const
  Wavefront OBJ **形式のデータを描画する手続き**.
- virtual void draw (const GgSimpleShader &shader, GLint first=0, GLsizei count=0) const
  Wavefront OBJ **形式のデータを描画する手続き**.

6.7.1 詳解

Wavefront OBJ 形式のファイル.

gg.h の 5955 行目に定義があります。

6.7.2 構築子と解体子

6.7.2.1 gg::GgObj::~GgObj ( ) [virtual]

デストラクタ.

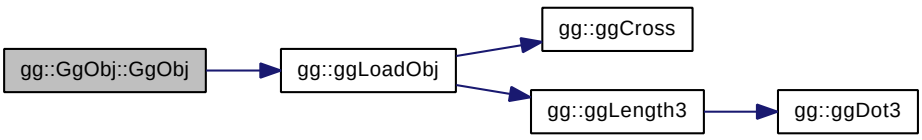
gg.cpp の 7936 行目に定義があります。

6.7.2.2 gg::GgObj::GgObj (const char * *name*, bool *normalize* = false)

コンストラクタ.

gg.cpp の 7949 行目に定義があります。

呼び出し関係図:



6.7.3 関数詳解

6.7.3.1 void gg::GgObj::draw (const GgSimpleShader * *shader* = NULL, GLint *first* = 0, GLsizei *count* = 0) const [virtual]

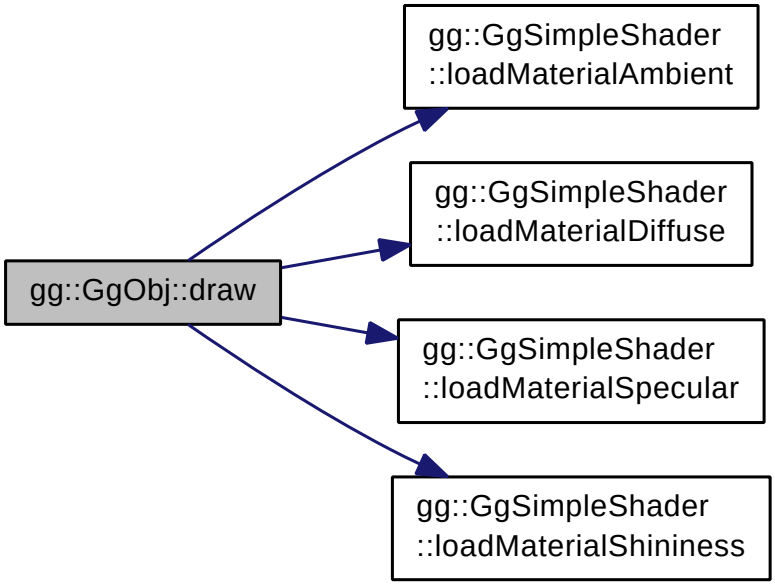
Wavefront OBJ 形式のデータを描画する手続き.

引数

shader	GgSimpleShader 型のシェーダのオブジェクトのポインタ.
first	描画する最初のパーツ番号.
count	描画するパーツの数.

gg.cpp の 7970 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:

6.7.3.2 virtual void gg::GgObj::draw (const GgSimpleShader & *shader*, GLint *first* = 0, GLsizei *count* = 0) const [inline], [virtual]

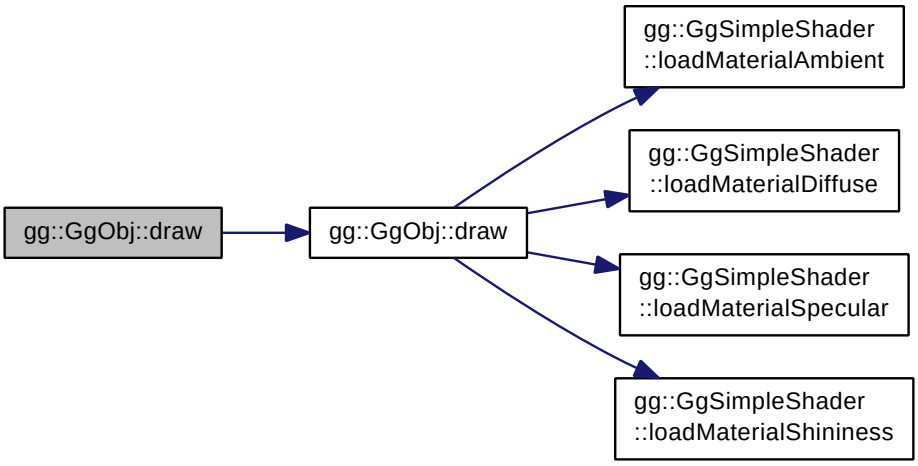
Wavefront OBJ 形式のデータを描画する手続き.

引数

shader	GgSimpleShader 型のシェーダのオブジェクト.
first	描画する最初のパーツ番号.
count	描画するパーツの数.

gg.h の 5990 行目に定義があります。

呼び出し関係図:



6.7.3.3 const GgTriangles* gg::GgObj::get ( ) const [inline]

形状データの取り出し.

戻り値

GgTriangles 型の形状データのポインタ.

gg.h の 5975 行目に定義があります。
このクラス詳解は次のファイルから抽出されました:

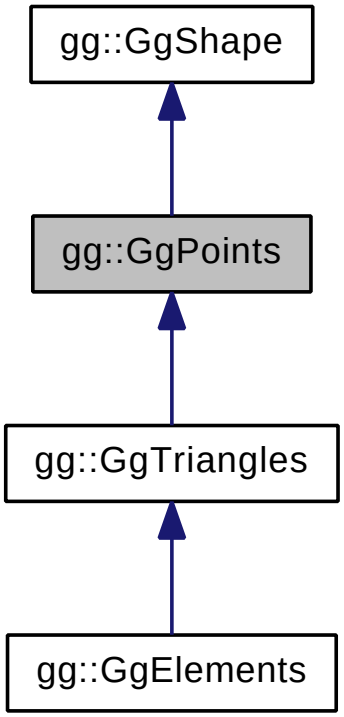
- gg.h
- gg.cpp

6.8 gg::GgPoints クラス

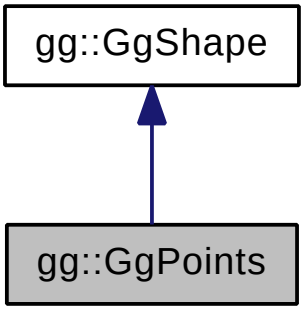
ポイント.

`#include <gg.h>`

gg::GgPoints の継承関係図



gg::GgPoints 連携図



公開メンバ関数

- virtual ~GgPoints ()
  デストラクタ.
- GgPoints (GLenum mode=GL_POINTS)
  コンストラクタ.
- GgPoints (GLuint nv, const GLfloat(*pos)[3], GLenum mode=GL_POINTS, GLenum usage=GL_STATIC_DRAW)
  コンストラクタ.
- GgPoints (const GgPoints &o)
  コピーコンストラクタ.
- GgPoints & operator= (const GgPoints &o)
- void send (GLuint nv, const GLfloat(*pos)[3], GLuint offset=0)
  既存のバッファオブジェクトに頂点の位置データを転送する.

- void load (GLuint nv, const GLfloat(∗pos)[3], GLenum usage=GL_STATIC_DRAW)
  バッファオブジェクトを確保して頂点の位置データを格納する.
- GLuint pbuf () const
  頂点の位置データを格納した頂点バッファオブジェクト名を取り出す.
- GLuint pnum () const
  データの数を取り出す.
- virtual void draw (GLint first=0, GLsizei count=0) const
  ポイントの描画.

6.8.1 詳解

ポイント.

gg.h の 5104 行目に定義があります。

6.8.2 構築子と解体子

6.8.2.1 virtual gg::GgPoints::∼GgPoints ( ) [inline], [virtual]

デストラクタ.

gg.h の 5127 行目に定義があります。

6.8.2.2 gg::GgPoints::GgPoints (GLenum *mode* = GL_POINTS) [inline]

コンストラクタ.

gg.h の 5130 行目に定義があります。

6.8.2.3 gg::GgPoints::GgPoints (GLuint *nv*, const GLfloat(∗) *pos[3]*, GLenum *mode* = GL_POINTS, GLenum *usage* = GL_STATIC_DRAW) [inline]

コンストラクタ.

引数

nv	頂点数.
pos	この図形の頂点の位置のデータの配列 (NULL ならデータを転送しない).
mode	描画する基本図形の種類.
usage	バッファオブジェクトの使い方.

gg.h の 5138 行目に定義があります。

6.8.2.4 gg::GgPoints::GgPoints (const GgPoints & *o*) [inline]

コピーコンストラクタ.

gg.h の 5145 行目に定義があります。

6.8.3 関数詳解

6.8.3.1 void gg::GgPoints::draw (GLint *first* = 0, GLsizei *count* = 0) const [virtual]

ポイントの描画.

gg::GgShapeを実装しています。

gg::GgElementsで再実装されています。

gg.cpp の 7526 行目に定義があります。

6.8.3.2 void gg::GgPoints::load (GLuint *nv*, const GLfloat(*) *pos[3]*, GLenum *usage* = GL_STATIC_DRAW) [inline]

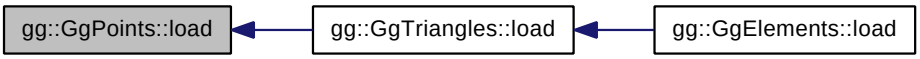
バッファオブジェクトを確保して頂点の位置データを格納する.

引数

nv	頂点のデータの数 (頂点数).
pos	頂点の位置データが格納されてている領域の先頭のポインタ.
usage	バッファオブジェクトの使い方.

gg.h の 5172 行目に定義があります。

被呼び出し関係図:



6.8.3.3 GgPoints& gg::GgPoints::operator= (const GgPoints & *o*) [inline]

gg.h の 5149 行目に定義があります。

呼び出し関係図:

被呼び出し関係図:

6.8.3.4 GLuint gg::GgPoints::pbuf ( ) const [inline]

頂点の位置データを格納した頂点バッファオブジェクト名を取り出す.

戻り値

この図形の頂点の位置データを格納した頂点バッファオブジェクト名.

gg.h の 5179 行目に定義があります。

呼び出し関係図:

6.8.3.5 GLuint gg::GgPoints::pnum ( ) const [inline]

データの数を取り出す.

戻り値

この図形の頂点の位置データの数 (頂点数).

gg.h の 5186 行目に定義があります。
呼び出し関係図:

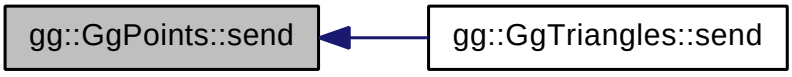
6.8.3.6 void gg::GgPoints::send (GLuint *nv*, const GLfloat(*) *pos[3]*, GLuint *offset* = 0) [inline]

既存のバッファオブジェクトに頂点の位置データを転送する.
引数

nv	転送する頂点の位置データの数 (0 ならバッファ全体).
pos	転送元の頂点の位置データが格納されてている領域の先頭のポインタ.
offset	転送先のバッファオブジェクトの先頭の要素番号.

gg.h の 5163 行目に定義があります。
呼び出し関係図:

被呼び出し関係図:



このクラス詳解は次のファイルから抽出されました:

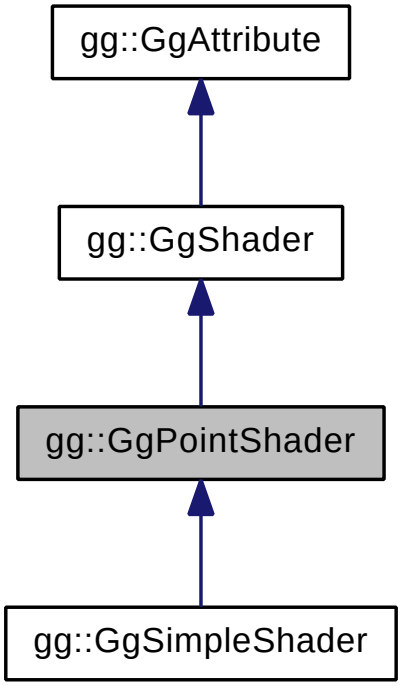
- gg.h
- gg.cpp

6.9 gg::GgPointShader クラス

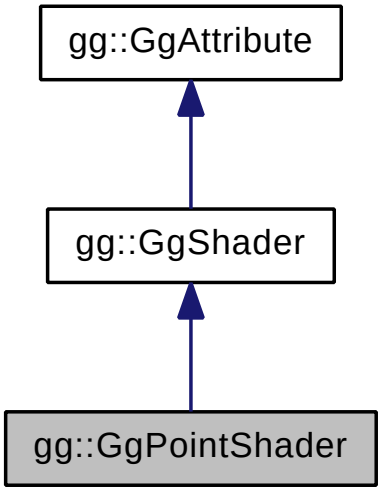
ポイントのシェーダ.

```
#include <gg.h>
```

gg::GgPointShader の継承関係図



gg::GgPointShader 連携図



公開メンバ関数

- virtual ~GgPointShader ()
  デストラクタ.
- GgPointShader ()
  コンストラクタ.
- GgPointShader (const char *vert, const char *frag=0, const char *geom=0, GLint nvarying=0, const char **varyings=0)
  コンストラクタ
- GgPointShader (const GgPointShader &o)

コピーコンストラクタ.

- GgPointShader & operator= (const GgPointShader &o)
- virtual void loadMatrix (const GgMatrix &mp, const GgMatrix &mv) const

  変換行列を設定する.
- virtual void loadMatrix (const GLfloat *mp, const GLfloat *mv) const

  変換行列を設定する.

その他の継承メンバ

6.9.1 詳解

ポイントのシェーダ.

gg.h の 5571 行目に定義があります。

6.9.2 構築子と解体子

6.9.2.1 virtual gg::GgPointShader::~GgPointShader ( ) [inline], [virtual]

デストラクタ.

gg.h の 5584 行目に定義があります。

6.9.2.2 gg::GgPointShader::GgPointShader ( ) [inline]

コンストラクタ.

gg.h の 5587 行目に定義があります。

6.9.2.3 gg::GgPointShader::GgPointShader (const char * *vert*, const char * *frag* = 0, const char * *geom* = 0, GLint *nvarying* = 0, const char ** *varyings* = 0) [inline]

コンストラクタ

引数

vert	バーテックスシェーダのソースファイル名.
frag	フラグメントシェーダのソースファイル名 (0 なら不使用).
geom	ジオメトリシェーダのソースファイル名 (0 なら不使用).
nvarying	フィードバックする varying 変数の数 (0 なら不使用).
varyings	フィードバックする varying 変数のリスト.

gg.h の 5595 行目に定義があります。

6.9.2.4 gg::GgPointShader::GgPointShader (const GgPointShader & *o*) [inline]

コピーコンストラクタ.

gg.h の 5608 行目に定義があります。

6.9.3 関数詳解

6.9.3.1 virtual void gg::GgPointShader::loadMatrix (const GgMatrix & *mp*, const GgMatrix & *mv*) const [inline], [virtual]

変換行列を設定する.

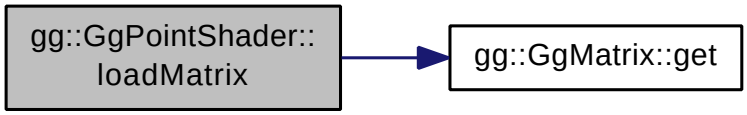
引数

mp	GgMatrix 型の投影変換行列.
mv	GgMatrix 型のモデルビュー変換行列.

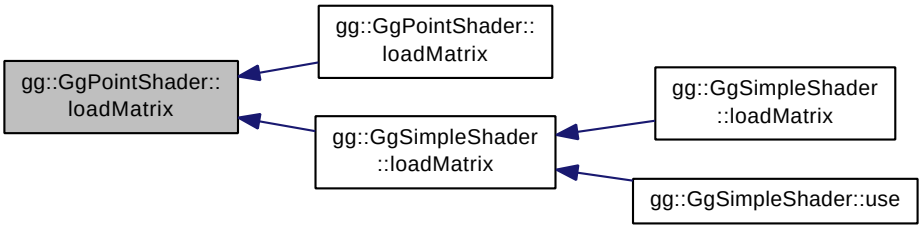
gg::GgSimpleShaderで再実装されています。

gg.h の 5625 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:



6.9.3.2 virtual void gg::GgPointShader::loadMatrix (const GLfloat * *mp*, const GLfloat * *mv*) const [inline], [virtual]

変換行列を設定する.

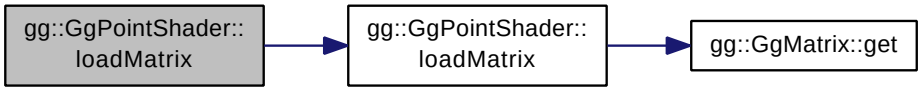
引数

mp	GLfloat 型の 16 要素の配列に格納された投影変換行列.
mv	GLfloat 型の 16 要素の配列に格納されたモデルビュー変換行列.

gg::GgSimpleShaderで再実装されています。

gg.h の 5635 行目に定義があります。

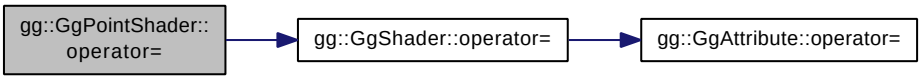
呼び出し関係図:



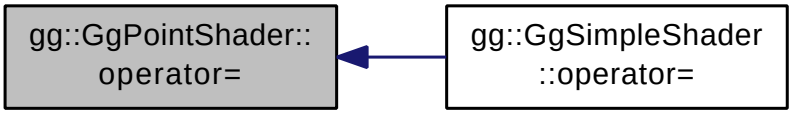
6.9.3.3 GgPointShader& gg::GgPointShader::operator= (const GgPointShader & *o*) [inline]

gg.h の 5612 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:



このクラス詳解は次のファイルから抽出されました:

- gg.h

6.10 gg::GgQuaternion クラス

四元数.

`#include <gg.h>`

公開メンバ関数

- ~GgQuaternion ()
  デストラクタ.
- GgQuaternion ()
  コンストラクタ.
- GgQuaternion (GLfloat x, GLfloat y, GLfloat z, GLfloat w)
  コンストラクタ.
- GgQuaternion (const GLfloat *a)
  コンストラクタ.
- GgQuaternion (const GgQuaternion &q)
  コピーコンストラクタ.
- GLfloat norm () const
  四元数のノルムを求める.
- GgQuaternion & load (GLfloat x, GLfloat y, GLfloat z, GLfloat w)
  四元数を格納する.
- GgQuaternion & load (const GLfloat *a)
  四元数を格納する.
- GgQuaternion & load (const GgQuaternion &q)
  四元数を格納する.
- GgQuaternion & loadAdd (GLfloat x, GLfloat y, GLfloat z, GLfloat w)
  四元数に別の四元数を加算した結果を格納する.
- GgQuaternion & loadAdd (const GLfloat *a)
  四元数に別の四元数を加算した結果を格納する.
- GgQuaternion & loadAdd (const GgQuaternion &q)
  四元数に別の四元数を加算した結果を格納する.
- GgQuaternion & loadSubtract (GLfloat x, GLfloat y, GLfloat z, GLfloat w)

四元数から別の四元数を減算した結果を格納する.

- GgQuaternion & loadSubtract (const GLfloat *a)

  四元数から別の四元数を減算した結果を格納する.
- GgQuaternion & loadSubtract (const GgQuaternion &q)

  四元数から別の四元数を減算した結果を格納する.
- GgQuaternion & loadMultiply (GLfloat x, GLfloat y, GLfloat z, GLfloat w)

  四元数に別の四元数を乗算した結果を格納する.
- GgQuaternion & loadMultiply (const GLfloat *a)

  四元数に別の四元数を乗算した結果を格納する.
- GgQuaternion & loadMultiply (const GgQuaternion &q)

  四元数に別の四元数を乗算した結果を格納する.
- GgQuaternion & loadDivide (GLfloat x, GLfloat y, GLfloat z, GLfloat w)

  四元を別の四元数で除算した結果を格納する.
- GgQuaternion & loadDivide (const GLfloat *a)

  四元を別の四元数で除算した結果を格納する.
- GgQuaternion & loadDivide (const GgQuaternion &q)

  四元を別の四元数で除算した結果を格納する.
- GgQuaternion add (GLfloat x, GLfloat y, GLfloat z, GLfloat w) const

  四元数に別の四元数を加算した結果を返す.
- GgQuaternion add (const GLfloat *a) const

  四元数に別の四元数を加算した結果を返す.
- GgQuaternion add (const GgQuaternion &q) const

  四元数に別の四元数を加算した結果を返す.
- GgQuaternion subtract (GLfloat x, GLfloat y, GLfloat z, GLfloat w) const

  四元数から別の四元数を減算した結果を返す.
- GgQuaternion subtract (const GLfloat *a) const

  四元数から別の四元数を減算した結果を返す.
- GgQuaternion subtract (const GgQuaternion &q) const

  四元数から別の四元数を減算した結果を返す.
- GgQuaternion multiply (GLfloat x, GLfloat y, GLfloat z, GLfloat w) const

  四元数に別の四元数を乗算した結果を返す.
- GgQuaternion multiply (const GLfloat *a) const

  四元数に別の四元数を乗算した結果を返す.
- GgQuaternion multiply (const GgQuaternion &q) const

  四元数に別の四元数を乗算した結果を返す.
- GgQuaternion divide (GLfloat x, GLfloat y, GLfloat z, GLfloat w) const

  四元数を別の四元数で除算した結果を返す.
- GgQuaternion divide (const GLfloat *a) const

  四元数を別の四元数で除算した結果を返す.
- GgQuaternion divide (const GgQuaternion &q) const

  四元数を別の四元数で除算した結果を返す.
- GgQuaternion & operator= (const GLfloat *a)
- GgQuaternion & operator= (const GgQuaternion &q)
- GgQuaternion & operator+= (const GLfloat *a)
- GgQuaternion & operator+= (const GgQuaternion &q)
- GgQuaternion & operator-= (const GLfloat *a)
- GgQuaternion & operator-= (const GgQuaternion &q)
- GgQuaternion & operator*= (const GLfloat *a)
- GgQuaternion & operator*= (const GgQuaternion &q)
- GgQuaternion & operator/= (const GLfloat *a)
- GgQuaternion & operator/= (const GgQuaternion &q)

- GgQuaternion operator+ (const GLfloat *a) const
- GgQuaternion operator+ (const GgQuaternion &q) const
- GgQuaternion operator- (const GLfloat *a) const
- GgQuaternion operator- (const GgQuaternion &q) const
- GgQuaternion operator* (const GLfloat *a) const
- GgQuaternion operator* (const GgQuaternion &q) const
- GgQuaternion operator/ (const GLfloat *a) const
- GgQuaternion operator/ (const GgQuaternion &q) const
- GgQuaternion & loadMatrix (const GLfloat *a)

  回転の変換行列を表す四元数を格納する.
- GgQuaternion & loadMatrix (const GgMatrix &m)

  回転の変換行列 *m* **を表す四元数を格納する**.
- GgQuaternion & loadIdentity ()

  単位元を格納する.
- GgQuaternion & loadRotate (GLfloat x, GLfloat y, GLfloat z, GLfloat a)

  (x, y, z) **を軸として角度** *a* **回転する四元数を格納する**.
- GgQuaternion & loadRotate (const GLfloat *v, GLfloat a)

  (v[0], v[1], v[2]) **を軸として角度** *a* **回転する四元数を格納する**.
- GgQuaternion & loadRotate (const GLfloat *v)

  (v[0], v[1], v[2]) **を軸として角度** *v[3]* **回転する四元数を格納する**.
- GgQuaternion & loadRotateX (GLfloat a)

  x **軸中心に角度** *a* **回転する四元数を格納する**.
- GgQuaternion & loadRotateY (GLfloat a)

  y **軸中心に角度** *a* **回転する四元数を格納する**.
- GgQuaternion & loadRotateZ (GLfloat a)

  z **軸中心に角度** *a* **回転する四元数を格納する**.
- GgQuaternion rotate (GLfloat x, GLfloat y, GLfloat z, GLfloat a) const

  四元数を *(x, y, z)* **を軸として角度** *a* **回転した四元数を返す**.
- GgQuaternion rotate (const GLfloat *v, GLfloat a) const

  四元数を *(v[0], v[1], v[2])* **を軸として角度** *a* **回転した四元数を返す**.
- GgQuaternion rotate (const GLfloat *v) const

  四元数を *(v[0], v[1], v[2])* **を軸として角度** *v[3]* **回転した四元数を返す**.
- GgQuaternion rotateX (GLfloat a) const

  四元数を *x* **軸中心に角度** *a* **回転した四元数を返す**.
- GgQuaternion rotateY (GLfloat a) const

  四元数を *y* **軸中心に角度** *a* **回転した四元数を返す**.
- GgQuaternion rotateZ (GLfloat a) const

  四元数を *z* **軸中心に角度** *a* **回転した四元数を返す**.
- GgQuaternion & loadEuler (GLfloat heading, GLfloat pitch, GLfloat roll)

  オイラー角 *(heading, pitch, roll)* **で与えられた回転を表す四元数を格納する**.
- GgQuaternion & loadEuler (const GLfloat *e)

  オイラー角 *(e[0], e[1], e[2])* **で与えられた回転を表す四元数を格納する**.
- GgQuaternion euler (GLfloat heading, GLfloat pitch, GLfloat roll) const

  四元数をオイラー角 *(heading, pitch, roll)* **で回転した四元数を返す**.
- GgQuaternion euler (const GLfloat *e) const

  四元数をオイラー角 *(e[0], e[1], e[2])* **で回転した四元数を返す**.
- GgQuaternion & loadSlerp (const GLfloat *a, const GLfloat *b, GLfloat t)

  球面線形補間の結果を格納する.
- GgQuaternion & loadSlerp (const GgQuaternion &q, const GgQuaternion &r, GLfloat t)

  球面線形補間の結果を格納する.
- GgQuaternion & loadSlerp (const GgQuaternion &q, const GLfloat *a, GLfloat t)

球面線形補間の結果を格納する.

- GgQuaternion & loadSlerp (const GLfloat *a, const GgQuaternion &q, GLfloat t)

  球面線形補間の結果を格納する.
- GgQuaternion & loadNormalize (const GLfloat *a)

  引数に指定した四元数を正規化して格納する.
- GgQuaternion & loadNormalize (const GgQuaternion &q)

  引数に指定した四元数を正規化して格納する.
- GgQuaternion & loadConjugate (const GLfloat *a)

  引数に指定した四元数の共役四元数を格納する.
- GgQuaternion & loadConjugate (const GgQuaternion &q)

  引数に指定した四元数の共役四元数を格納する.
- GgQuaternion & loadInvert (const GLfloat *a)

  引数に指定した四元数の逆元を格納する.
- GgQuaternion & loadInvert (const GgQuaternion &q)

  引数に指定した四元数の逆元を格納する.
- GgQuaternion slerp (GLfloat *a, GLfloat t) const

  球面線形補間の結果を返す.
- GgQuaternion slerp (const GgQuaternion &q, GLfloat t) const

  球面線形補間の結果を返す.
- GgQuaternion normalize () const

  正規化する.
- GgQuaternion conjugate () const

  共役四元数に変換する.
- GgQuaternion invert () const

  逆元に変換する.
- const GLfloat * get () const

  四元数を取り出す.
- void get (GLfloat *a) const

  四元数を取り出す.
- void getMatrix (GLfloat *a) const

  四元数が表す回転の変換行列を *a* に求める.
- void getMatrix (GgMatrix &m) const

  四元数が表す回転の変換行列を *m* に求める.
- GgMatrix getMatrix () const

  四元数が表す回転の変換行列を取り出す.
- void getConjugateMatrix (GLfloat *a) const

  四元数の共役が表す回転の変換行列を *a* に求める.
- void getConjugateMatrix (GgMatrix &m) const

  四元数の共役が表す回転の変換行列を *m* に求める.
- GgMatrix getConjugateMatrix () const

  四元数の共役が表す回転の変換行列を取り出す.

6.10.1 詳解

四元数.

gg.h の 3674 行目に定義があります。

6.10.2 構築子と解体子

6.10.2.1 gg::GgQuaternion::~GgQuaternion ( ) [inline]

デストラクタ.

gg.h の 3694 行目に定義があります。

6.10.2.2 gg::GgQuaternion::GgQuaternion ( ) [inline]

コンストラクタ.

gg.h の 3697 行目に定義があります。

6.10.2.3 gg::GgQuaternion::GgQuaternion (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *w*) [inline]

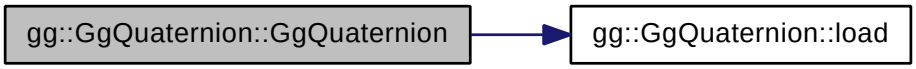
コンストラクタ.

引数

x	四元数の x 要素.
y	四元数の y 要素.
z	四元数の z 要素.
w	四元数の w 要素.

gg.h の 3704 行目に定義があります。

呼び出し関係図:



6.10.2.4 gg::GgQuaternion::GgQuaternion (const GLfloat * *a*) [inline]

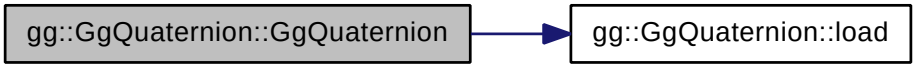
コンストラクタ.

引数

a	GLfloat 型の GLfloat 型の 4 要素の配列に格納した四元数.

gg.h の 3711 行目に定義があります。

呼び出し関係図:



6.10.2.5 gg::GgQuaternion::GgQuaternion (const GgQuaternion & *q*) [inline]

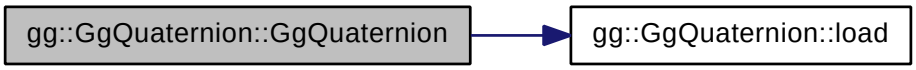
コピーコンストラクタ.

引数

q	GgQuaternion 型の四元数.

gg.h の 3718 行目に定義があります。

呼び出し関係図:



6.10.3 関数詳解

6.10.3.1 GgQuaternion gg::GgQuaternion::add (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *w*) const [inline]

四元数に別の四元数を加算した結果を返す.

引数

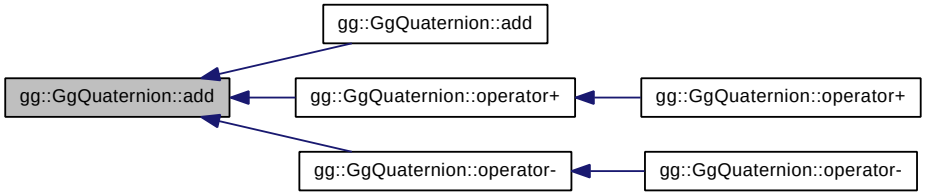
x	加える四元数の x 要素.
y	加える四元数の y 要素.
z	加える四元数の z 要素.
w	加える四元数の w 要素.

戻り値

(x, y, z, w) を加えた四元数.

gg.h の 3885 行目に定義があります。

被呼び出し関係図:



6.10.3.2 GgQuaternion gg::GgQuaternion::add (const GLfloat * *a*) const [inline]

四元数に別の四元数を加算した結果を返す.

引数

a	GLfloat 型の GLfloat 型の 4 要素の配列に格納した四元数.

戻り値

a を加えた四元数.

gg.h の 3898 行目に定義があります。

呼び出し関係図:

6.10.3.3 GgQuaternion gg::GgQuaternion::add (const GgQuaternion & *q*) const [inline]

四元数に別の四元数を加算した結果を返す.

引数

q	GgQuaternion 型の四元数.

戻り値

q を加えた四元数.

gg.h の 3906 行目に定義があります。

呼び出し関係図:

6.10.3.4 GgQuaternion gg::GgQuaternion::conjugate ( ) const [inline]

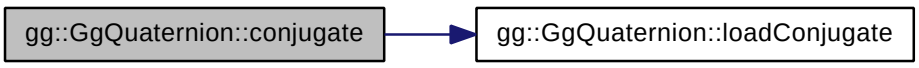
共役四元数に変換する.

戻り値

共役四元数.

gg.h の 4342 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:

6.10.3.5 GgQuaternion gg::GgQuaternion::divide (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *w*) const [inline]

四元数を別の四元数で除算した結果を返す.

引数

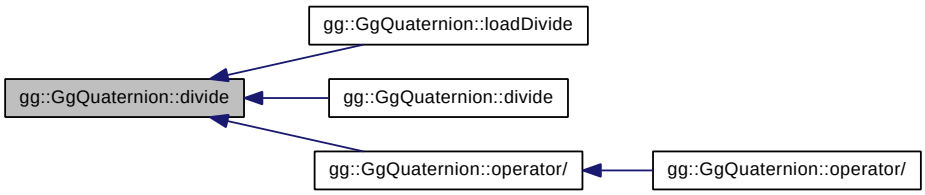
x	割る四元数の x 要素.
y	割る四元数の y 要素.
z	割る四元数の z 要素.
w	割る四元数の w 要素.

戻り値

(x, y, z, w) を割った四元数.

gg.h の 3979 行目に定義があります。

被呼び出し関係図:



6.10.3.6 GgQuaternion gg::GgQuaternion::divide (const GLfloat * *a*) const [inline]

四元数を別の四元数で除算した結果を返す.

引数

a	GLfloat 型の GLfloat 型の 4 要素の配列に格納した四元数.

戻り値

a で割った四元数.

gg.h の 3988 行目に定義があります。

呼び出し関係図:

6.10.3.7 GgQuaternion gg::GgQuaternion::divide (const GgQuaternion & *q*) const [inline]

四元数を別の四元数で除算した結果を返す.

引数

q	GgQuaternion 型の四元数.

戻り値

q で割った四元数.

gg.h の 3999 行目に定義があります。

呼び出し関係図:

6.10.3.8 GgQuaternion gg::GgQuaternion::euler (GLfloat *heading*, GLfloat *pitch*, GLfloat *roll*) const [inline]

四元数をオイラー角 (heading, pitch, roll) で回転した四元数を返す.

引数

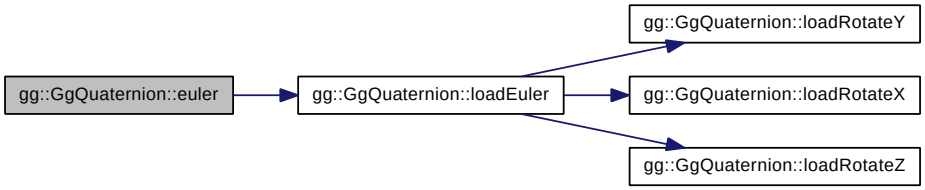
heading	y 軸中心の回転角.
pitch	x 軸中心の回転角.
roll	z 軸中心の回転角.

戻り値

回転した四元数.

gg.h の 4215 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:

6.10.3.9 GgQuaternion gg::GgQuaternion::euler (const GLfloat * *e*) const [inline]

四元数をオイラー角 (e[0], e[1], e[2]) で回転した四元数を返す.

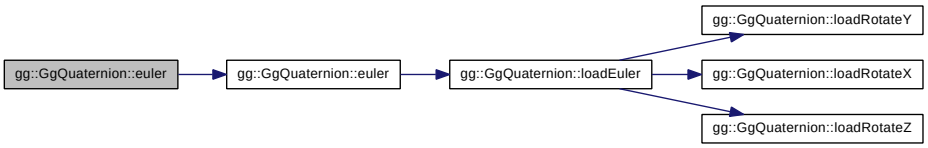
引数

e	オイラー角を表す GLfloat 型の 3 要素の配列 (heading, pitch, roll).

戻り値

回転した四元数.

gg.h の 4224 行目に定義があります。
呼び出し関係図:



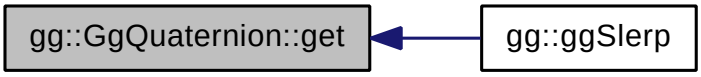
6.10.3.10 const GLfloat* gg::GgQuaternion::get ( ) const [inline]

四元数を取り出す.

戻り値

四元数を表す GLfloat 型の 4 要素の配列.

gg.h の 4360 行目に定義があります。
被呼び出し関係図:



6.10.3.11 void gg::GgQuaternion::get (GLfloat * a) const [inline]

四元数を取り出す.
引数

a	四元数を格納する GLfloat 型の 4 要素の配列.

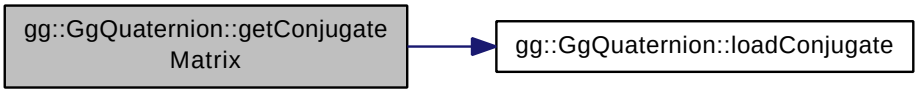
gg.h の 4367 行目に定義があります。

6.10.3.12 void gg::GgQuaternion::getConjugateMatrix (GLfloat * a) const [inline]

四元数の共役が表す回転の変換行列を a に求める.
引数

a	回転の変換行列を格納する GLfloat 型の 16 要素の配列.

gg.h の 4400 行目に定義があります。
呼び出し関係図:



6.10.3.13 void gg::GgQuaternion::getConjugateMatrix (GgMatrix & *m*) const [inline]

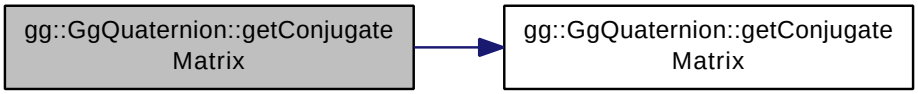
四元数の共役が表す回転の変換行列を m に求める.

引数

m	回転の変換行列を格納する GgMatrix 型の変数.

gg.h の 4409 行目に定義があります。

呼び出し関係図:



6.10.3.14 GgMatrix gg::GgQuaternion::getConjugateMatrix ( ) const [inline]

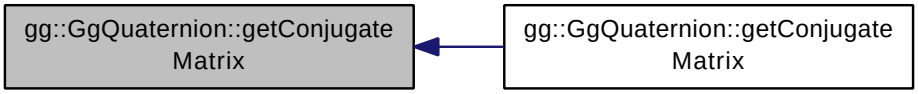
四元数の共役が表す回転の変換行列を取り出す.

戻り値

回転の変換を表す GgMatrix 型の変換行列.

gg.h の 4416 行目に定義があります。

被呼び出し関係図:



6.10.3.15 void gg::GgQuaternion::getMatrix (GLfloat * *a*) const [inline]

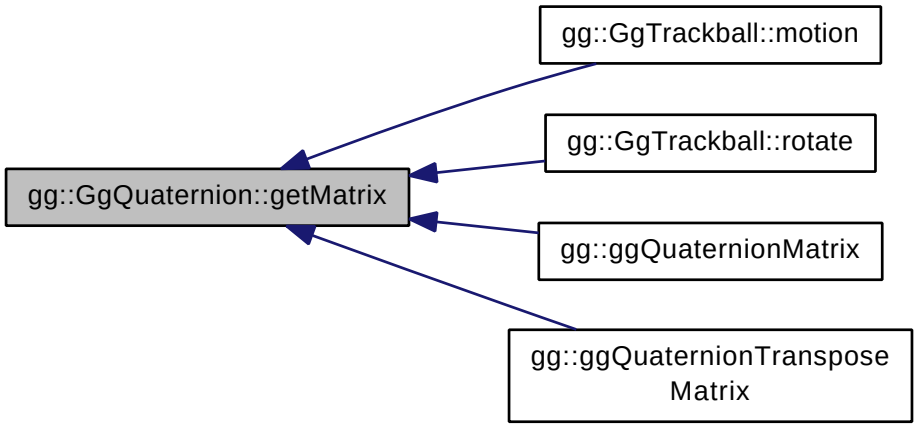
四元数が表す回転の変換行列を a に求める.

引数

a	回転の変換行列を格納する GLfloat 型の 16 要素の配列.

gg.h の 4377 行目に定義があります。

被呼び出し関係図:



6.10.3.16 void gg::GgQuaternion::getMatrix (GgMatrix & *m*) const [inline]

四元数が表す回転の変換行列を m に求める.

引数

m	回転の変換行列を格納する GgMatrix 型の変数.

gg.h の 4384 行目に定義があります。

呼び出し関係図:

6.10.3.17 GgMatrix gg::GgQuaternion::getMatrix ( ) const [inline]

四元数が表す回転の変換行列を取り出す.

戻り値

回転の変換を表す GgMatrix 型の変換行列.

gg.h の 4391 行目に定義があります。

被呼び出し関係図:

6.10.3.18 GgQuaternion gg::GgQuaternion::invert ( ) const [inline]

逆元に変換する.

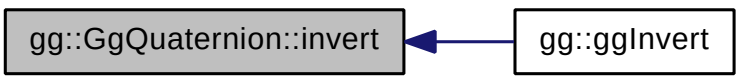
戻り値

四元数の逆元.

gg.h の 4351 行目に定義があります。

呼び出し関係図:

被呼び出し関係図:



6.10.3.19 GgQuaternion& gg::GgQuaternion::load (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *w*) [inline]

四元数を格納する.

引数

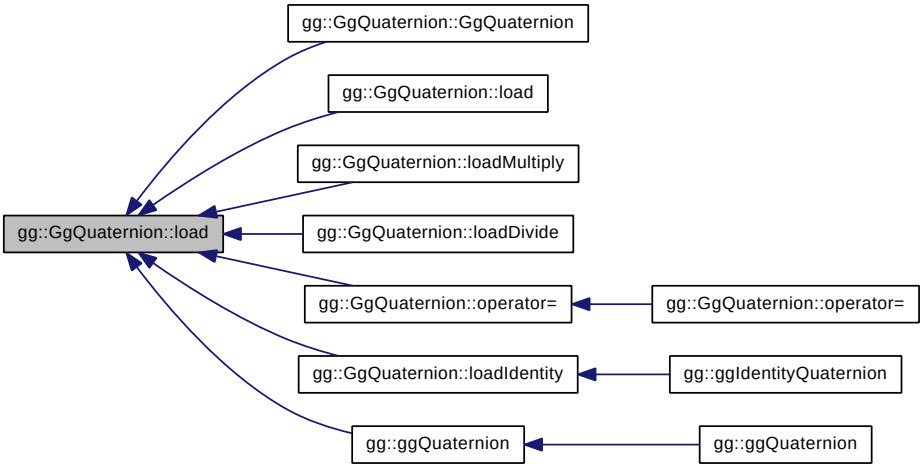
x	四元数の x 要素.
y	四元数の y 要素.
z	四元数の z 要素.
w	四元数の w 要素.

戻り値

設定した四元数.

gg.h の 3736 行目に定義があります。

被呼び出し関係図:



6.10.3.20 GgQuaternion& gg::GgQuaternion::load (const GLfloat * *a*) [inline]

四元数を格納する.

引数

a	GLfloat 型の GLfloat 型の 4 要素の配列に格納した四元数.

戻り値

設定した四元数.

gg.h の 3748 行目に定義があります。

呼び出し関係図:

6.10.3.21 GgQuaternion& gg::GgQuaternion::load (const GgQuaternion & *q*) [inline]

四元数を格納する.

引数

q	GgQuaternion 型の四元数.

戻り値

設定した四元数.

gg.h の 3756 行目に定義があります。

呼び出し関係図:

6.10.3.22 GgQuaternion& gg::GgQuaternion::loadAdd (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *w*) [inline]

四元数に別の四元数を加算した結果を格納する.

引数

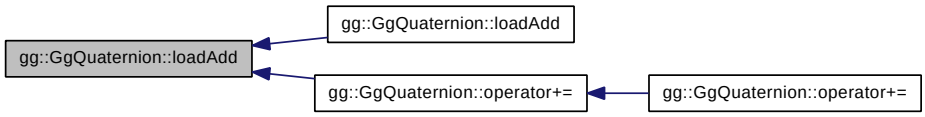
x	加える四元数の x 要素.
y	加える四元数の y 要素.
z	加える四元数の z 要素.
w	加える四元数の w 要素.

戻り値

(x, y, z, w) を加えた四元数.

gg.h の 3767 行目に定義があります。

被呼び出し関係図:



6.10.3.23 GgQuaternion& gg::GgQuaternion::loadAdd (const GLfloat * *a*) [inline]

四元数に別の四元数を加算した結果を格納する.

引数

a	GLfloat 型の GLfloat 型の 4 要素の配列に格納した四元数.

戻り値

a を加えた四元数.

gg.h の 3779 行目に定義があります。

呼び出し関係図:

6.10.3.24 GgQuaternion& gg::GgQuaternion::loadAdd (const GgQuaternion & *q*) [inline]

四元数に別の四元数を加算した結果を格納する.

引数

q	GgQuaternion 型の四元数.

戻り値

q を加えた四元数.

gg.h の 3787 行目に定義があります。

呼び出し関係図:

6.10.3.25 gg::GgQuaternion & gg::GgQuaternion::loadConjugate (const GLfloat * *a*)

引数に指定した四元数の共役四元数を格納する.

引数

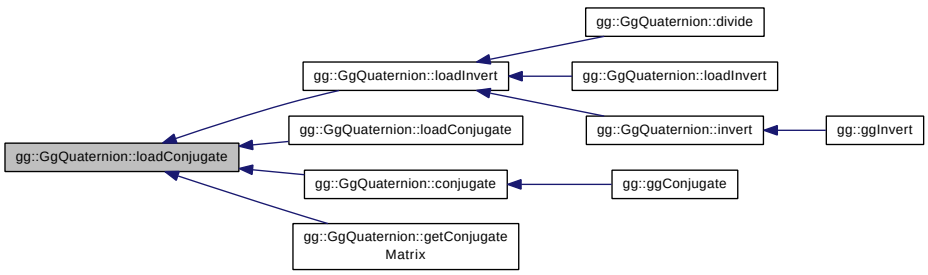
a	四元数を格納した GLfloat 型の 4 要素の配列.

戻り値

共役四元数.

gg.cpp の 7377 行目に定義があります。

被呼び出し関係図:



6.10.3.26 GgQuaternion& gg::GgQuaternion::loadConjugate (const GgQuaternion & *q*) [inline]

引数に指定した四元数の共役四元数を格納する.

引数

q	GgQuaternion 型の四元数.

戻り値

共役四元数.

gg.h の 4291 行目に定義があります。

呼び出し関係図:

6.10.3.27 GgQuaternion& gg::GgQuaternion::loadDivide (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *w*) [inline]

四元を別の四元数で除算した結果を格納する.

引数

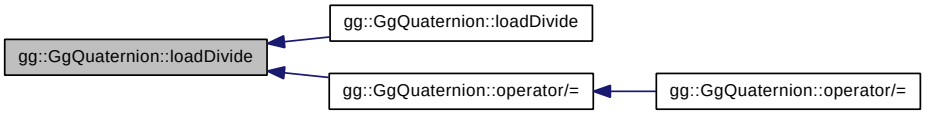
x	割る四元数の x 要素.
y	割る四元数の y 要素.
z	割る四元数の z 要素.
w	割る四元数の w 要素.

戻り値

(x, y, z, w) を割った四元数.

gg.h の 3857 行目に定義があります。

被呼び出し関係図:



6.10.3.28 GgQuaternion& gg::GgQuaternion::loadDivide (const GLfloat * *a*) [inline]

四元を別の四元数で除算した結果を格納する.

引数

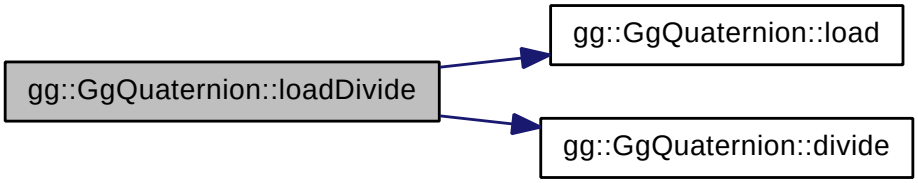
a	GLfloat 型の GLfloat 型の 4 要素の配列に格納した四元数.

戻り値

a で割った四元数.

gg.h の 3866 行目に定義があります。

呼び出し関係図:



6.10.3.29 GgQuaternion& gg::GgQuaternion::loadDivide (const GgQuaternion & *q*) [inline]

四元を別の四元数で除算した結果を格納する.

引数

q	GgQuaternion 型の四元数.

戻り値

q で割った四元数.

gg.h の 3874 行目に定義があります。

呼び出し関係図:

6.10.3.30 gg::GgQuaternion & gg::GgQuaternion::loadEuler (GLfloat *heading*, GLfloat *pitch*, GLfloat *roll*)

オイラー角 (heading, pitch, roll) で与えられた回転を表す四元数を格納する.

引数

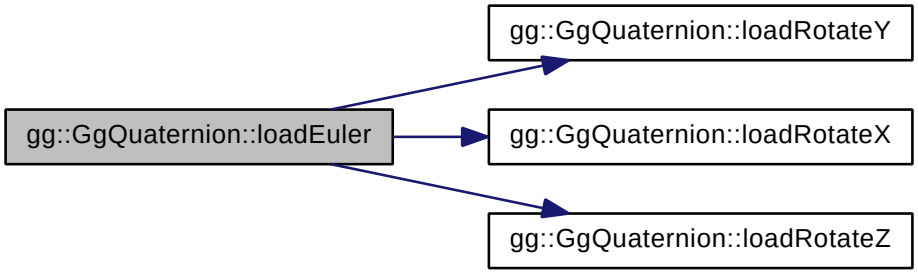
heading	y 軸中心の回転角.
pitch	x 軸中心の回転角.
roll	z 軸中心の回転角.

戻り値

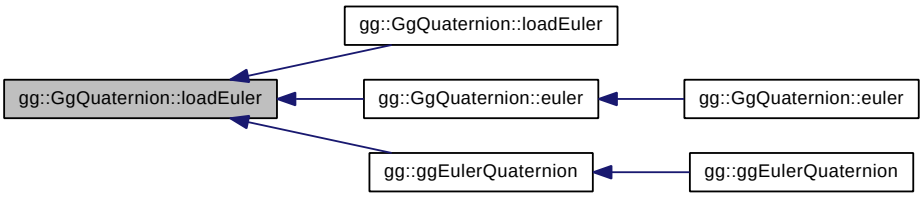
格納した回転を表す四元数.

gg.cpp の 7341 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:



6.10.3.31 GgQuaternion& gg::GgQuaternion::loadEuler (const GLfloat * e) [inline]

オイラー角 (e[0], e[1], e[2]) で与えられた回転を表す四元数を格納する.

引数

e	オイラー角を表す GLfloat 型の 3 要素の配列 (heading, pitch, roll).

戻り値

格納した回転を表す四元数.

gg.h の 4205 行目に定義があります。

呼び出し関係図:

6.10.3.32 GgQuaternion& gg::GgQuaternion::loadIdentity ( ) [inline]

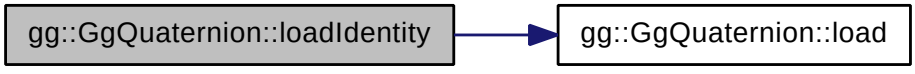
単位元を格納する.

戻り値

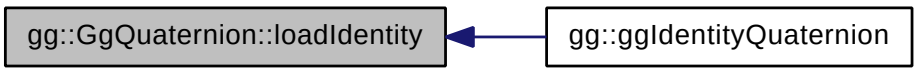
格納された単位元.

gg.h の 4097 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:



6.10.3.33 gg::GgQuaternion & gg::GgQuaternion::loadInvert (const GLfloat * *a*)

引数に指定した四元数の逆元を格納する.

引数

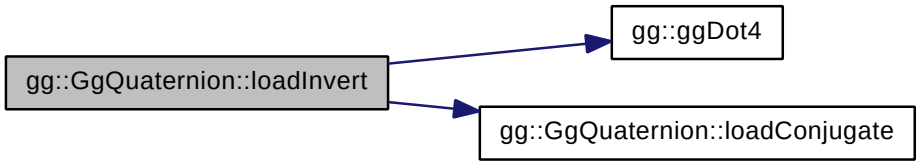
a	四元数を格納した GLfloat 型の 4 要素の配列.

戻り値

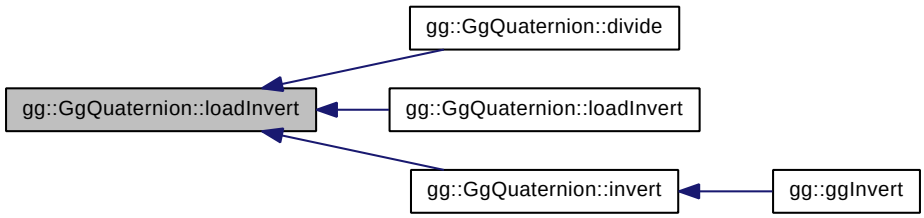
四元数の逆元.

gg.cpp の 7391 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:



6.10.3.34 GgQuaternion& gg::GgQuaternion::loadInvert (const GgQuaternion & *q*) [inline]

引数に指定した四元数の逆元を格納する.

引数

q	GgQuaternion 型の四元数.

戻り値

四元数の逆元.

gg.h の 4304 行目に定義があります。

呼び出し関係図:

6.10.3.35 GgQuaternion& gg::GgQuaternion::loadMatrix (const GLfloat * *a*) [inline]

回転の変換行列を表す四元数を格納する.

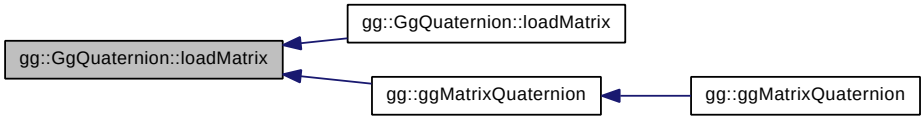
引数

a	GLfloat 型の 16 要素の変換行列.

戻り値

a による回転の変換に相当する四元数.

gg.h の 4081 行目に定義があります。
被呼び出し関係図:



6.10.3.36 GgQuaternion& gg::GgQuaternion::loadMatrix (const GgMatrix & *m*) [inline]

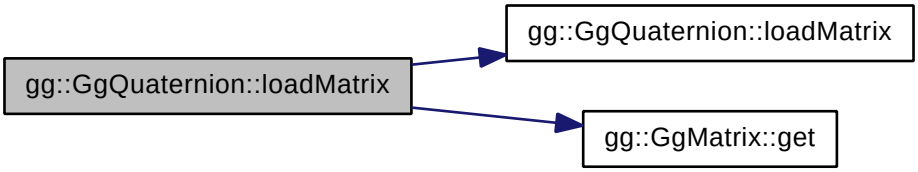
回転の変換行列 m を表す四元数を格納する.
引数

m	Ggmatrix 型の変換行列.

戻り値

m による回転の変換に相当する四元数.

gg.h の 4090 行目に定義があります。
呼び出し関係図:



6.10.3.37 GgQuaternion& gg::GgQuaternion::loadMultiply (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *w*) [inline]

四元数に別の四元数を乗算した結果を格納する.
引数

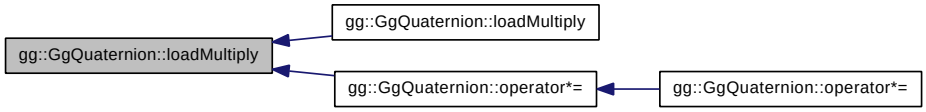
x	掛ける四元数の x 要素.
y	掛ける四元数の y 要素.
z	掛ける四元数の z 要素.
w	掛ける四元数の w 要素.

戻り値

(x, y, z, w) を掛けた四元数.

gg.h の 3829 行目に定義があります。

被呼び出し関係図:



6.10.3.38 GgQuaternion& gg::GgQuaternion::loadMultiply (const GLfloat * a) [inline]

四元数に別の四元数を乗算した結果を格納する.

引数

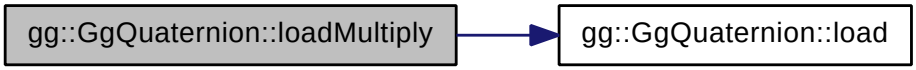
a	GLfloat 型の GLfloat 型の 4 要素の配列に格納した四元数.

戻り値

a を乗じた四元数.

gg.h の 3838 行目に定義があります。

呼び出し関係図:



6.10.3.39 GgQuaternion& gg::GgQuaternion::loadMultiply (const GgQuaternion & q) [inline]

四元数に別の四元数を乗算した結果を格納する.

引数

q	GgQuaternion 型の四元数.

戻り値

q を乗じた四元数.

gg.h の 3846 行目に定義があります。

呼び出し関係図:

6.10.3.40 gg::GgQuaternion & gg::GgQuaternion::loadNormalize (const GLfloat * a)

引数に指定した四元数を正規化して格納する.

引数

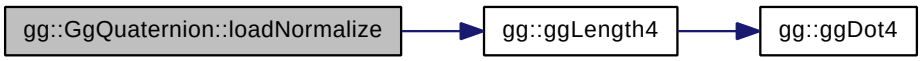
a	四元数を格納した GLfloat 型の 4 要素の配列.

戻り値

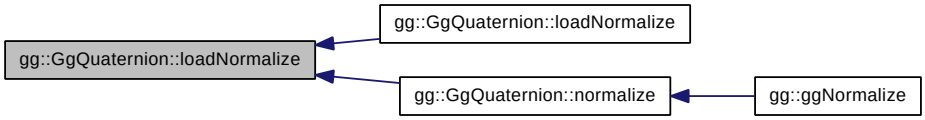
正規化された四元数.

gg.cpp の 7357 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:



6.10.3.41 GgQuaternion& gg::GgQuaternion::loadNormalize (const GgQuaternion & *q*) [inline]

引数に指定した四元数を正規化して格納する.

引数

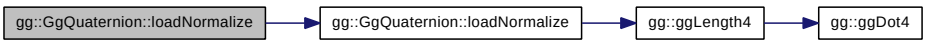
q	GgQuaternion 型の四元数.

戻り値

正規化された四元数.

gg.h の 4278 行目に定義があります。

呼び出し関係図:



6.10.3.42 gg::GgQuaternion & gg::GgQuaternion::loadRotate (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *a*)

(x, y, z) を軸として角度 a 回転する四元数を格納する.

引数

x	軸ベクトルの x 成分.
y	軸ベクトルの y 成分.

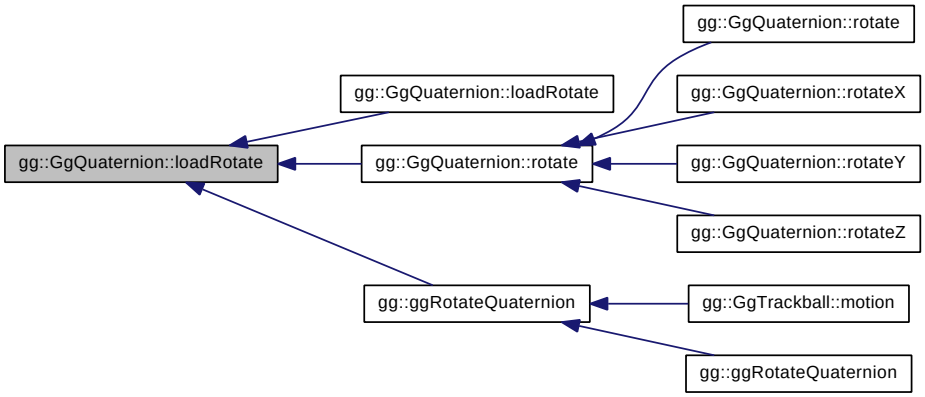
z	軸ベクトルの z 成分.
a	回転角.

戻り値

格納された回転を表す四元数.

gg.cpp の 7275 行目に定義があります。

被呼び出し関係図:



6.10.3.43 GgQuaternion& gg::GgQuaternion::loadRotate (const GLfloat * *v*, GLfloat *a*) [inline]

(v[0], v[1], v[2]) を軸として角度 a 回転する四元数を格納する.

引数

v	軸ベクトルを表す GLfloat 型の 3 要素の配列.
a	回転角.

戻り値

格納された回転を表す四元数.

gg.h の 4114 行目に定義があります。

呼び出し関係図:

6.10.3.44 GgQuaternion& gg::GgQuaternion::loadRotate (const GLfloat * *v*) [inline]

(v[0], v[1], v[2]) を軸として角度 v[3] 回転する四元数を格納する.

引数

v	軸ベクトルと回転角を格納した GLfloat 型の 4 要素の配列.

戻り値

格納された回転を表す四元数.

gg.h の 4122 行目に定義があります。

呼び出し関係図:

6.10.3.45 gg::GgQuaternion & gg::GgQuaternion::loadRotateX (GLfloat *a*)

x 軸中心に角度 a 回転する四元数を格納する.

引数

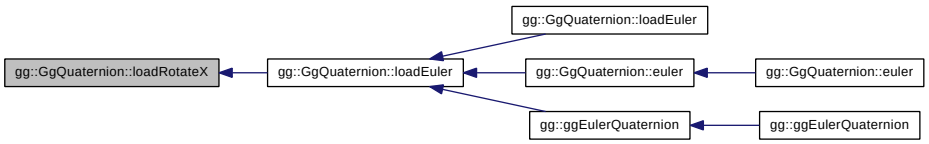
a	回転角.

戻り値

格納された回転を表す四元数.

gg.cpp の 7299 行目に定義があります。

被呼び出し関係図:



6.10.3.46 gg::GgQuaternion & gg::GgQuaternion::loadRotateY (GLfloat *a*)

y 軸中心に角度 a 回転する四元数を格納する.

引数

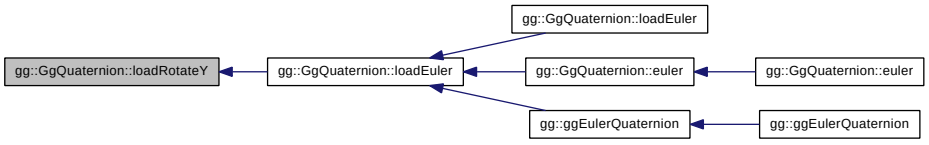
a	回転角.

戻り値

格納された回転を表す四元数.

gg.cpp の 7313 行目に定義があります。

被呼び出し関係図:



6.10.3.47 gg::GgQuaternion & gg::GgQuaternion::loadRotateZ (GLfloat *a*)

z 軸中心に角度 a 回転する四元数を格納する.

引数

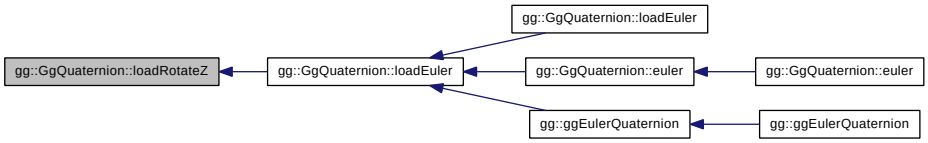
v	軸ベクトルを表す GLfloat 型の 4 要素の配列.

戻り値

格納された回転を表す四元数.

gg.cpp の 7327 行目に定義があります。

被呼び出し関係図:



6.10.3.48 GgQuaternion& gg::GgQuaternion::loadSlerp (const GLfloat * *a*, const GLfloat * *b*, GLfloat *t*) [inline]

球面線形補間の結果を格納する.

引数

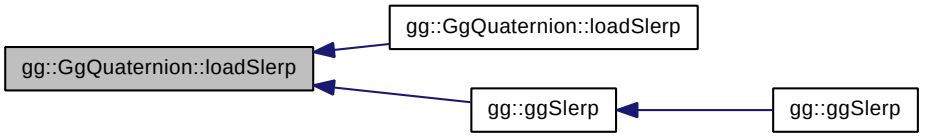
a	四元数を格納した GLfloat 型の 4 要素の配列.
b	四元数を格納した GLfloat 型の 4 要素の配列.
t	補間パラメータ.

戻り値

格納した a, b を t で内分した四元数.

gg.h の 4234 行目に定義があります。

被呼び出し関係図:



6.10.3.49 GgQuaternion& gg::GgQuaternion::loadSlerp (const GgQuaternion & *q*, const GgQuaternion & *r*, GLfloat *t*) [inline]

球面線形補間の結果を格納する.

引数

q	GgQuaternion 型の四元数.
r	GgQuaternion 型の四元数.
t	補間パラメータ.

戻り値

格納した q, r を t で内分した四元数.

gg.h の 4245 行目に定義があります。

呼び出し関係図:

6.10.3.50 GgQuaternion& gg::GgQuaternion::loadSlerp (const GgQuaternion & *q*, const GLfloat * *a*, GLfloat *t*) [inline]

球面線形補間の結果を格納する.

引数

q	GgQuaternion 型の四元数.
a	四元数を格納した GLfloat 型の 4 要素の配列.
t	補間パラメータ.

戻り値

格納した q, a を t で内分した四元数.

gg.h の 4255 行目に定義があります。

呼び出し関係図:

6.10.3.51 GgQuaternion& gg::GgQuaternion::loadSlerp (const GLfloat * *a*, const GgQuaternion & *q*, GLfloat *t*) [inline]

球面線形補間の結果を格納する.

引数

a	四元数を格納した GLfloat 型の 4 要素の配列.

q	GgQuaternion 型の四元数.
t	補間パラメータ.

戻り値

格納した a, q を t で内分した四元数.

gg.h の 4265 行目に定義があります。

呼び出し関係図:

6.10.3.52 GgQuaternion& gg::GgQuaternion::loadSubtract (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *w*) [inline]

四元数から別の四元数を減算した結果を格納する.

引数

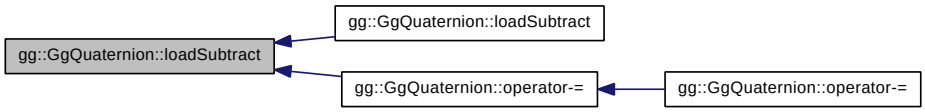
x	引く四元数の x 要素.
y	引く四元数の y 要素.
z	引く四元数の z 要素.
w	引く四元数の w 要素.

戻り値

(x, y, z, w) を引いた四元数.

gg.h の 3798 行目に定義があります。

被呼び出し関係図:



6.10.3.53 GgQuaternion& gg::GgQuaternion::loadSubtract (const GLfloat * *a*) [inline]

四元数から別の四元数を減算した結果を格納する.

引数

a	GLfloat 型の GLfloat 型の 4 要素の配列に格納した四元数.

戻り値

a を引いた四元数.

gg.h の 3810 行目に定義があります。

呼び出し関係図:

6.10.3.54 GgQuaternion& gg::GgQuaternion::loadSubtract (const GgQuaternion & *q*) [inline]

四元数から別の四元数を減算した結果を格納する.

引数

q	GgQuaternion 型の四元数.

戻り値

q を引いた四元数.

gg.h の 3818 行目に定義があります。

呼び出し関係図:

6.10.3.55 GgQuaternion gg::GgQuaternion::multiply (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *w*) const [inline]

四元数に別の四元数を乗算した結果を返す.

引数

x	掛ける四元数の x 要素.
y	掛ける四元数の y 要素.
z	掛ける四元数の z 要素.
w	掛ける四元数の w 要素.

戻り値

(x, y, z, w) を掛けた四元数.

gg.h の 3949 行目に定義があります。

6.10.3.56 GgQuaternion gg::GgQuaternion::multiply (const GLfloat * *a*) const [inline]

四元数に別の四元数を乗算した結果を返す.

引数

a	GLfloat 型の GLfloat 型の 4 要素の配列に格納した四元数.

戻り値

a を掛けた四元数.

gg.h の 3958 行目に定義があります。

6.10.3.57 GgQuaternion gg::GgQuaternion::multiply (const GgQuaternion & *q*) const [inline]

四元数に別の四元数を乗算した結果を返す.

引数

q	GgQuaternion 型の四元数.

戻り値

q を掛けた四元数.

gg.h の 3968 行目に定義があります。

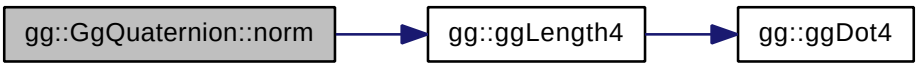
6.10.3.58 GLfloat gg::GgQuaternion::norm ( ) const [inline]

四元数のノルムを求める.

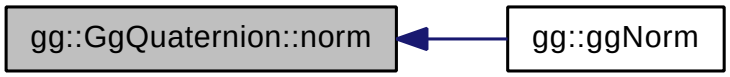
戻り値

四元数のノルム.

gg.h の 3725 行目に定義があります。
呼び出し関係図:



被呼び出し関係図:



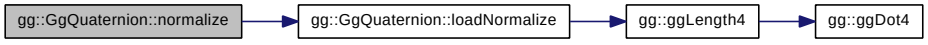
6.10.3.59 GgQuaternion gg::GgQuaternion::normalize ( ) const [inline]

正規化する.

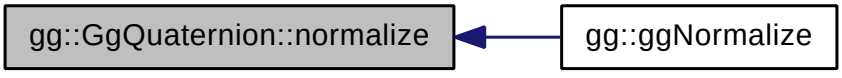
戻り値

正規化された四元数.

gg.h の 4333 行目に定義があります。
呼び出し関係図:



被呼び出し関係図:



6.10.3.60 GgQuaternion gg::GgQuaternion::operator∗ (const GLfloat ∗ *a*) const [inline]

gg.h の 4061 行目に定義があります。
被呼び出し関係図:

6.10.3.61 GgQuaternion gg::GgQuaternion::operator∗ (const GgQuaternion & *q*) const [inline]

gg.h の 4065 行目に定義があります。
呼び出し関係図:

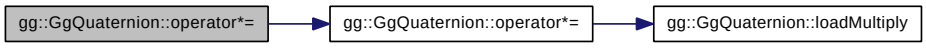
6.10.3.62 GgQuaternion& gg::GgQuaternion::operator∗= (const GLfloat ∗ *a*) [inline]

gg.h の 4029 行目に定義があります。
呼び出し関係図:

被呼び出し関係図:

6.10.3.63 GgQuaternion& gg::GgQuaternion::operator∗= (const GgQuaternion & *q*) [inline]

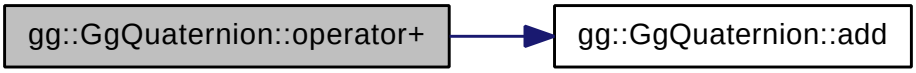
gg.h の 4033 行目に定義があります。
呼び出し関係図:



6.10.3.64 GgQuaternion gg::GgQuaternion::operator+ (const GLfloat ∗ *a*) const [inline]

gg.h の 4045 行目に定義があります。

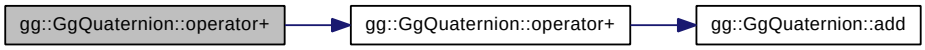
呼び出し関係図:



被呼び出し関係図:

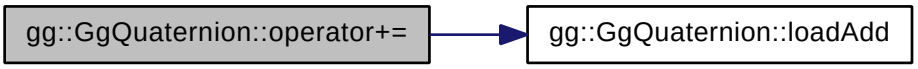
6.10.3.65 GgQuaternion gg::GgQuaternion::operator+ (const GgQuaternion & *q*) const [inline]

gg.h の 4049 行目に定義があります。
呼び出し関係図:



6.10.3.66 GgQuaternion& gg::GgQuaternion::operator+= (const GLfloat * *a*) [inline]

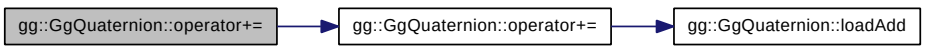
gg.h の 4013 行目に定義があります。
呼び出し関係図:



被呼び出し関係図:

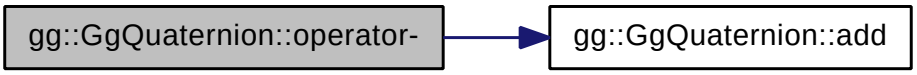
6.10.3.67 GgQuaternion& gg::GgQuaternion::operator+= (const GgQuaternion & *q*) [inline]

gg.h の 4017 行目に定義があります。
呼び出し関係図:



6.10.3.68 GgQuaternion gg::GgQuaternion::operator- (const GLfloat * *a*) const [inline]

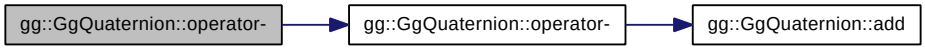
gg.h の 4053 行目に定義があります。
呼び出し関係図:



被呼び出し関係図:

6.10.3.69 GgQuaternion gg::GgQuaternion::operator- (const GgQuaternion & *q*) const [inline]

gg.h の 4057 行目に定義があります。
呼び出し関係図:



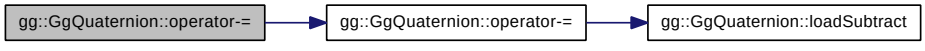
6.10.3.70 GgQuaternion& gg::GgQuaternion::operator-= (const GLfloat * *a*) [inline]

gg.h の 4021 行目に定義があります。
呼び出し関係図:

被呼び出し関係図:

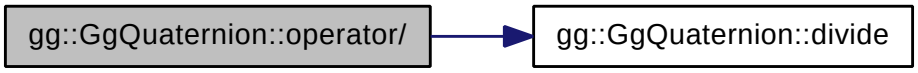
6.10.3.71 GgQuaternion& gg::GgQuaternion::operator-= (const GgQuaternion & *q*) [inline]

gg.h の 4025 行目に定義があります。
呼び出し関係図:



6.10.3.72 GgQuaternion gg::GgQuaternion::operator/ (const GLfloat * a) const [inline]

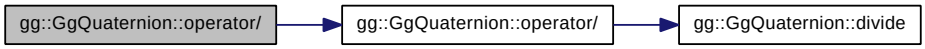
gg.h の 4069 行目に定義があります。
呼び出し関係図:



被呼び出し関係図:

6.10.3.73 GgQuaternion gg::GgQuaternion::operator/ (const GgQuaternion & q) const [inline]

gg.h の 4073 行目に定義があります。
呼び出し関係図:



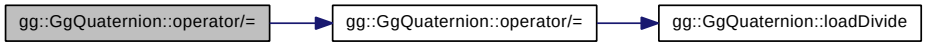
6.10.3.74 GgQuaternion& gg::GgQuaternion::operator/= (const GLfloat * a) [inline]

gg.h の 4037 行目に定義があります。
呼び出し関係図:

被呼び出し関係図:

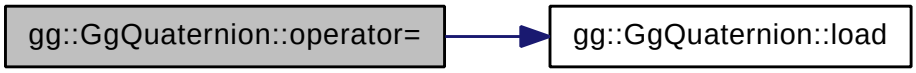
6.10.3.75 GgQuaternion& gg::GgQuaternion::operator/= (const GgQuaternion & q) [inline]

gg.h の 4041 行目に定義があります。
呼び出し関係図:



6.10.3.76 GgQuaternion& gg::GgQuaternion::operator= (const GLfloat * a) [inline]

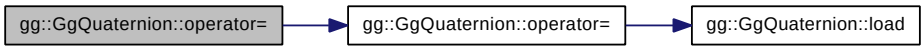
gg.h の 4005 行目に定義があります。
呼び出し関係図:



被呼び出し関係図:

6.10.3.77 GgQuaternion& gg::GgQuaternion::operator= (const GgQuaternion & q) [inline]

gg.h の 4009 行目に定義があります。
呼び出し関係図:



6.10.3.78 GgQuaternion gg::GgQuaternion::rotate (GLfloat x, GLfloat y, GLfloat z, GLfloat a) const [inline]

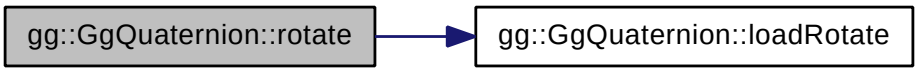
四元数を (x, y, z) を軸として角度 a 回転した四元数を返す.
引数

x	軸ベクトルの x 成分.
y	軸ベクトルの y 成分.
z	軸ベクトルの z 成分.
a	回転角.

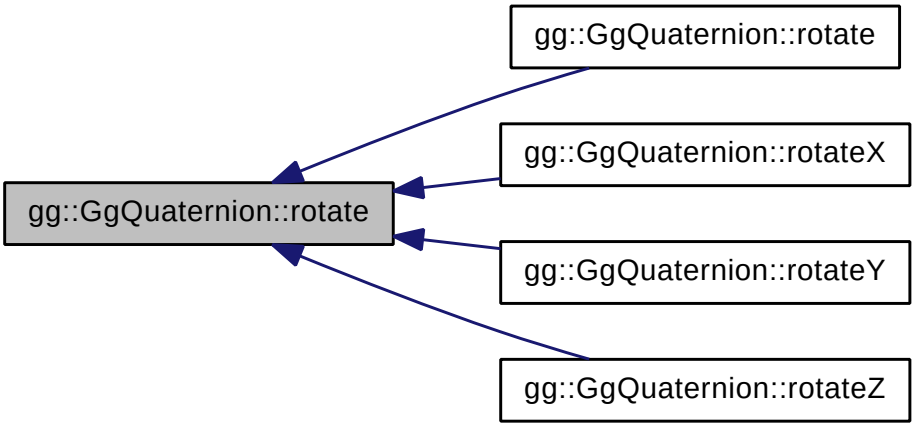
戻り値

回転した四元数.

gg.h の 4148 行目に定義があります。
呼び出し関係図:



被呼び出し関係図:



6.10.3.79 GgQuaternion gg::GgQuaternion::rotate (const GLfloat * *v*, GLfloat *a*) const [inline]

四元数を (v[0], v[1], v[2]) を軸として角度 a 回転した四元数を返す.

引数

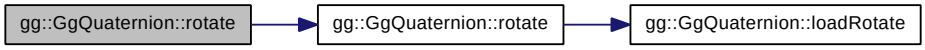
v	軸ベクトルを表す GLfloat 型の 3 要素の配列.
a	回転角.

戻り値

回転した四元数.

gg.h の 4158 行目に定義があります。

呼び出し関係図:



6.10.3.80 GgQuaternion gg::GgQuaternion::rotate (const GLfloat * *v*) const [inline]

四元数を (v[0], v[1], v[2]) を軸として角度 v[3] 回転した四元数を返す.

引数

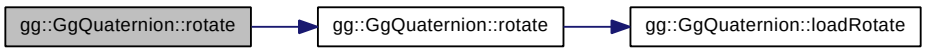
v	軸ベクトルを表す GLfloat 型の 4 要素の配列.

戻り値

回転した四元数.

gg.h の 4166 行目に定義があります。

呼び出し関係図:



6.10.3.81 GgQuaternion gg::GgQuaternion::rotateX (GLfloat *a*) const [inline]

四元数を x 軸中心に角度 a 回転した四元数を返す.

引数

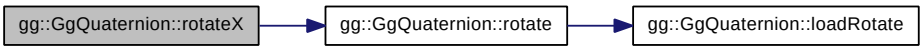
a	回転角.

戻り値

回転した四元数.

gg.h の 4174 行目に定義があります。

呼び出し関係図:



6.10.3.82 GgQuaternion gg::GgQuaternion::rotateY (GLfloat *a*) const [inline]

四元数を y 軸中心に角度 a 回転した四元数を返す.

引数

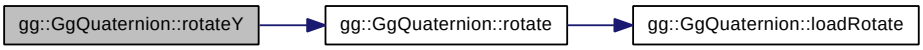
a	回転角.

戻り値

回転した四元数.

gg.h の 4182 行目に定義があります。

呼び出し関係図:



6.10.3.83 GgQuaternion gg::GgQuaternion::rotateZ (GLfloat *a*) const [inline]

四元数を z 軸中心に角度 a 回転した四元数を返す.

引数

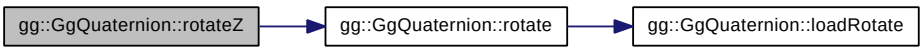
a	回転角.

戻り値

回転した四元数.

gg.h の 4190 行目に定義があります。

呼び出し関係図:



6.10.3.84 GgQuaternion gg::GgQuaternion::slerp (GLfloat * *a*, GLfloat *t*) const [inline]

球面線形補間の結果を返す.

引数

a	四元数を格納した GLfloat 型の 4 要素の配列.
t	補間パラメータ.

戻り値

四元数を a に対して t で内分した結果.

gg.h の 4313 行目に定義があります。

6.10.3.85 GgQuaternion gg::GgQuaternion::slerp (const GgQuaternion & *q*, GLfloat *t*) const [inline]

球面線形補間の結果を返す.

引数

q	GgQuaternion 型の四元数.
t	補間パラメータ.

戻り値

四元数を q に対して t で内分した結果.

gg.h の 4324 行目に定義があります。

6.10.3.86 GgQuaternion gg::GgQuaternion::subtract (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *w*) const [inline]

四元数から別の四元数を減算した結果を返す.

引数

x	引く四元数の x 要素.
y	引く四元数の y 要素.
z	引く四元数の z 要素.
w	引く四元数の w 要素.

戻り値

(x, y, z, w) を引いた四元数.

gg.h の 3917 行目に定義があります。

被呼び出し関係図:

6.10.3.87 GgQuaternion gg::GgQuaternion::subtract (const GLfloat * *a*) const [inline]

四元数から別の四元数を減算した結果を返す.

引数

a	GLfloat 型の GLfloat 型の 4 要素の配列に格納した四元数.

戻り値

a を引いた四元数.

gg.h の 3930 行目に定義があります。

呼び出し関係図:

6.10.3.88 GgQuaternion gg::GgQuaternion::subtract (const GgQuaternion & *q*) const [inline]

四元数から別の四元数を減算した結果を返す.

引数

q	GgQuaternion 型の四元数.

戻り値

q を引いた四元数.

gg.h の 3938 行目に定義があります。

呼び出し関係図:

このクラス詳解は次のファイルから抽出されました:

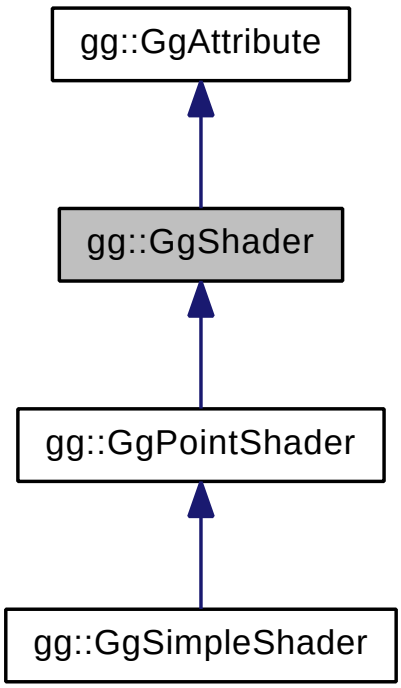
- gg.h
- gg.cpp

6.11 gg::GgShader クラス

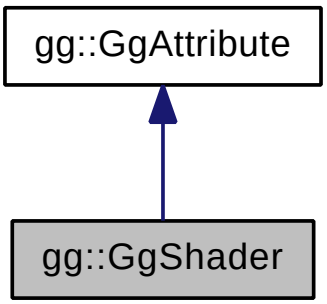
シェーダの基底クラス.

`#include <gg.h>`

gg::GgShader の継承関係図



gg::GgShader 連携図



公開メンバ関数

- virtual ~GgShader ()
  デストラクタ.
- GgShader ()
  コンストラクタ.
- GgShader (const char *vert, const char *frag=0, const char *geom=0, int nvarying=0, const char **varyings=0)
  コンストラクタ.
- GgShader (const GgShader &o)
  コピーコンストラクタ.
- GgShader & operator= (const GgShader &o)
- void setProgram (GLuint newProgram)
  別のシェーダのプログラムオブジェクトを登録する.

- void load (const char *vert, const char *frag=0, const char *geom=0, GLint nvarying=0, const char **varyings=0)
  シェーダのソースプログラムを読み込んでプログラムオブジェクトをする.
- void use () const
  シェーダプログラムの使用を開始する.
- void unuse () const
  シェーダプログラムの使用を終了する.
- GLuint get () const
  シェーダのプログラム名を得る.

その他の継承メンバ

6.11.1 詳解

シェーダの基底クラス.

シェーダのクラスはこのクラスを派生して作る.

gg.h の 5476 行目に定義があります。

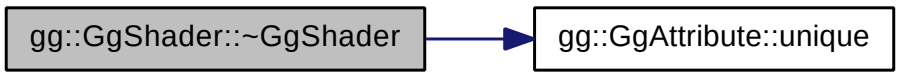
6.11.2 構築子と解体子

6.11.2.1 virtual gg::GgShader::~GgShader ( ) [inline], [virtual]

デストラクタ.

gg.h の 5485 行目に定義があります。

呼び出し関係図:



6.11.2.2 gg::GgShader::GgShader ( ) [inline]

コンストラクタ.

gg.h の 5496 行目に定義があります。

6.11.2.3 gg::GgShader::GgShader (const char * *vert*, const char * *frag* = 0, const char * *geom* = 0, int *nvarying* = 0, const char ** *varyings* = 0) [inline]

コンストラクタ.

引数

vert	バーテックスシェーダのソースファイル名.
frag	フラグメントシェーダのソースファイル名 (0 なら不使用).
geom	ジオメトリシェーダのソースファイル名 (0 なら不使用).

nvarying	フィードバックする varying 変数の数 (0 なら不使用).
varyings	フィードバックする varying 変数のリスト.

gg.h の 5505 行目に定義があります。

6.11.2.4 gg::GgShader::GgShader (const GgShader & *o*) [inline]

コピーコンストラクタ.

gg.h の 5510 行目に定義があります。

6.11.3 関数詳解

6.11.3.1 GLuint gg::GgShader::get ( ) const [inline]

シェーダのプログラム名を得る.

戻り値

シェーダのプログラム名.

gg.h の 5562 行目に定義があります。

6.11.3.2 void gg::GgShader::load (const char * *vert*, const char * *frag* = 0, const char * *geom* = 0, GLint *nvarying* = 0, const char ** *varyings* = 0) [inline]

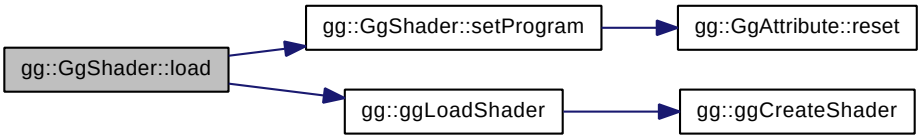
シェーダのソースプログラムを読み込んでプログラムオブジェクトをする.

引数

vert	バーテックスシェーダのソースファイル名.
frag	フラグメントシェーダのソースファイル名 (0 なら不使用).
geom	ジオメトリシェーダのソースファイル名 (0 なら不使用).
nvarying	フィードバックする varying 変数の数 (0 なら不使用).
varyings	フィードバックする varying 変数のリスト.

gg.h の 5542 行目に定義があります。

呼び出し関係図:

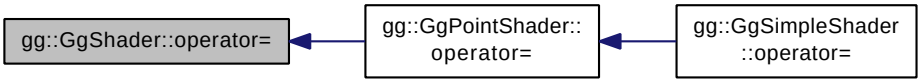


6.11.3.3 GgShader& gg::GgShader::operator= (const GgShader & *o*) [inline]

gg.h の 5514 行目に定義があります。

呼び出し関係図:

被呼び出し関係図:



6.11.3.4 void gg::GgShader::setProgram (GLuint *newProgram*) [inline]

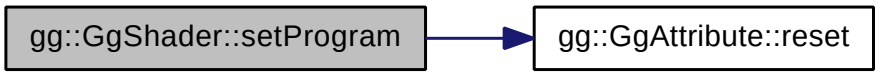
別のシェーダのプログラムオブジェクトを登録する.

引数

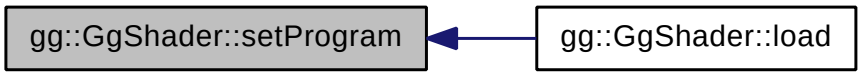
newProgram	別に作成したシェーダのプログラム名.

gg.h の 5526 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:



6.11.3.5 void gg::GgShader::unuse ( ) const [inline]

シェーダプログラムの使用を終了する.

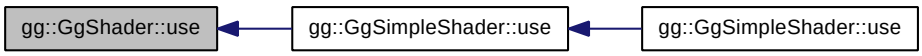
gg.h の 5555 行目に定義があります。

6.11.3.6 void gg::GgShader::use ( ) const [inline]

シェーダプログラムの使用を開始する.

gg.h の 5549 行目に定義があります。

被呼び出し関係図:



このクラス詳解は次のファイルから抽出されました:

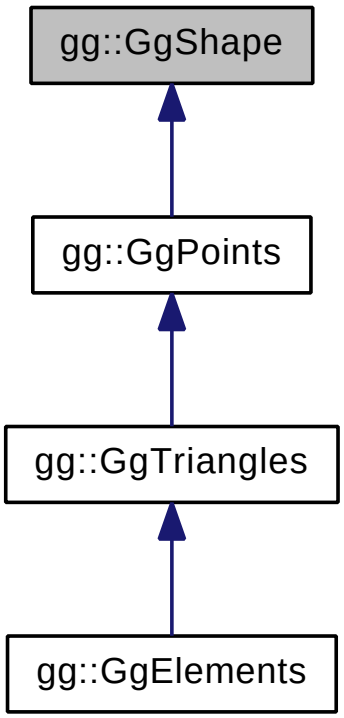
- gg.h

6.12 gg::GgShape クラス

形状データの基底クラス.

#include <gg.h>

gg::GgShape の継承関係図



公開メンバ関数

- virtual ~GgShape ()
  デストラクタ.
- GgShape (GLenum mode=0)
  コンストラクタ.
- GgShape (const GgShape &o)
  コピーコンストラクタ.
- GgShape & operator= (const GgShape &o)
- void use () const
  この図形の頂点配列オブジェクトを指定する.
- GLuint get () const
  頂点配列オブジェクト名を取り出す.
- void setMode (GLenum mode)
  基本図形の設定.
- GLenum getMode () const
  基本図形の検査.
- virtual void draw (GLint first=0, GLsizei count=0) const =0
  図形の描画.

6.12.1 詳解

形状データの基底クラス.

形状データのクラスはこのクラスを派生して作る. 基本図形の種類と頂点配列オブジェクトを保持する.

gg.h の 5024 行目に定義があります。

6.12.2 構築子と解体子

6.12.2.1 virtual gg::GgShape::~GgShape ( ) [inline], [virtual]

デストラクタ.

gg.h の 5035 行目に定義があります。

6.12.2.2 gg::GgShape::GgShape (GLenum *mode* = 0) [inline]

コンストラクタ.

引数

mode	基本図形の種類.

gg.h の 5043 行目に定義があります。

6.12.2.3 gg::GgShape::GgShape (const GgShape & *o*) [inline]

コピーコンストラクタ.

gg.h の 5051 行目に定義があります。

6.12.3 関数詳解

6.12.3.1 virtual void gg::GgShape::draw (GLint *first* = 0, GLsizei *count* = 0) const [pure virtual]

図形の描画.

この形状を描画する手続きをオーバーライドする.

gg::GgElements, gg::GgPointsで実装されています。

6.12.3.2 GLuint gg::GgShape::get ( ) const [inline]

頂点配列オブジェクト名を取り出す.

戻り値

頂点配列オブジェクト名.

gg.h の 5077 行目に定義があります。

6.12.3.3 GLenum gg::GgShape::getMode ( ) const [inline]

基本図形の検査.

戻り値

この頂点配列オブジェクトの基本図形の種類.

gg.h の 5091 行目に定義があります。

6.12.3.4 GgShape& gg::GgShape::operator= (const GgShape & *o*) [inline]

gg.h の 5058 行目に定義があります。
被呼び出し関係図:

6.12.3.5 void gg::GgShape::setMode (GLenum *mode*) [inline]

基本図形の設定.
引数

mode	基本図形の種類.

gg.h の 5084 行目に定義があります。

6.12.3.6 void gg::GgShape::use ( ) const [inline]

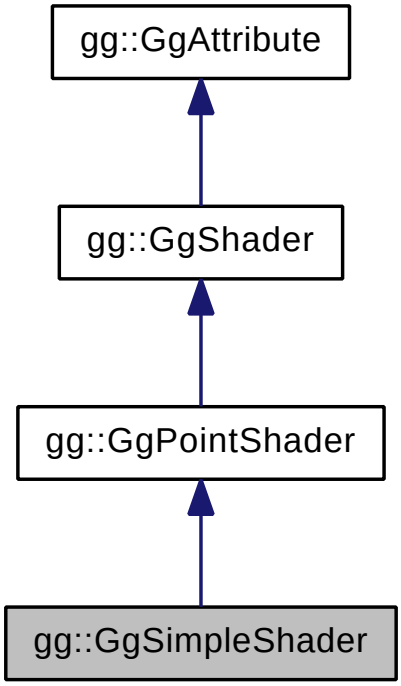
この図形の頂点配列オブジェクトを指定する.
gg.h の 5070 行目に定義があります。
このクラス詳解は次のファイルから抽出されました:

- gg.h

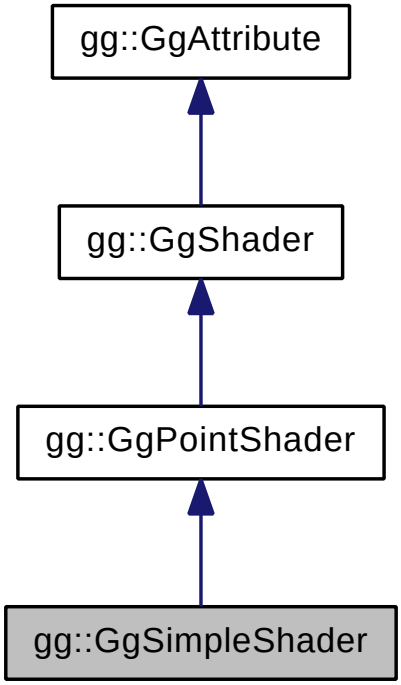
6.13 gg::GgSimpleShader クラス

三角形に単純な陰影付けを行うシェーダ.
`#include <gg.h>`

gg::GgSimpleShader の継承関係図



gg::GgSimpleShader 連携図



クラス

- struct Light
  光源の特性
- struct Material

材質の特性

公開メンバ関数

- virtual ~GgSimpleShader ()
  デストラクタ.
- GgSimpleShader ()
  コンストラクタ.
- GgSimpleShader (const char *vert, const char *frag=0, const char *geom=0, GLint nvarying=0, const char **varyings=0)
  コンストラクタ
- GgSimpleShader (const GgSimpleShader &o)
  コピーコンストラクタ.
- GgSimpleShader & operator= (const GgSimpleShader &o)
- void loadMaterialAmbient (GLfloat r, GLfloat g, GLfloat b, GLfloat a=1.0f) const
  環境光に対する反射係数を設定する.
- void loadMaterialAmbient (const GLfloat *amb) const
  環境光に対する反射係数を設定する.
- void loadMaterialDiffuse (GLfloat r, GLfloat g, GLfloat b, GLfloat a=1.0f) const
  拡散反射係数を設定する.
- void loadMaterialDiffuse (const GLfloat *diff) const
  拡散反射係数を設定する.
- void loadMaterialSpecular (GLfloat r, GLfloat g, GLfloat b, GLfloat a=1.0f) const
  鏡面反射係数を設定する.
- void loadMaterialSpecular (const GLfloat *spec) const
  鏡面反射係数を設定する.
- void loadMaterialShininess (GLfloat shi) const
  輝き係数を設定する.
- void loadMaterial (const Material &material) const
  材質の特性を設定する.
- void loadLightAmbient (GLfloat r, GLfloat g, GLfloat b, GLfloat a=1.0f) const
  光源の強度の環境光成分を設定する.
- void loadLightAmbient (const GLfloat *amb) const
  光源の強度の環境光成分を設定する.
- void loadLightDiffuse (GLfloat r, GLfloat g, GLfloat b, GLfloat a=1.0f) const
  光源の強度の拡散反射光成分を設定する.
- void loadLightDiffuse (const GLfloat *diff) const
  光源の強度の拡散反射光成分を設定する.
- void loadLightSpecular (GLfloat r, GLfloat g, GLfloat b, GLfloat a=1.0f) const
  光源の強度の鏡面反射光成分を設定する.
- void loadLightSpecular (const GLfloat *spec) const
  光源の強度の鏡面反射光成分を設定する.
- void loadLightPosition (GLfloat x, GLfloat y, GLfloat z, GLfloat w=1.0f) const
  光源の位置を設定する.
- void loadLightPosition (const GLfloat *pos) const
  光源の位置を設定する.
- void loadLightMaterial (const Light &light) const
  光源の特性を設定する.
- void loadLight (const Light &light) const
  光源の位置と特性を設定する.
- virtual void loadMatrix (const GgMatrix &mp, const GgMatrix &mv, const GgMatrix &mn) const

変換行列を設定する.

- virtual void loadMatrix (const GLfloat *mp, const GLfloat *mv, const GLfloat *mn) const
  変換行列を設定する.
- virtual void loadMatrix (const GgMatrix &mp, const GgMatrix &mv) const
  変換行列を設定する.
- virtual void loadMatrix (const GLfloat *mp, const GLfloat *mv) const
  変換行列を設定する.
- void use () const
  シェーダプログラムの使用を開始する.
- void use (const Light &light, const GgMatrix &mp, const GgMatrix &mv, const GgMatrix &mn) const
  光源と変換行列を指定してシェーダプログラムの使用を開始する.
- void use (const Light &light, const GLfloat *mp, const GLfloat *mv, const GLfloat *mn) const
  光源と変換行列を指定してシェーダプログラムの使用を開始する.
- void use (const Light &light, const GgMatrix &mp, const GgMatrix &mv) const
  光源と変換行列を指定してシェーダプログラムの使用を開始する.
- void use (const Light &light, const GLfloat *mp, const GLfloat *mv) const
  光源と変換行列を指定してシェーダプログラムの使用を開始する.

その他の継承メンバ

6.13.1 詳解

三角形に単純な陰影付けを行うシェーダ.

gg.h の 5644 行目に定義があります。

6.13.2 構築子と解体子

6.13.2.1 virtual gg::GgSimpleShader::~GgSimpleShader ( ) [inline], [virtual]

デストラクタ.

gg.h の 5664 行目に定義があります。

6.13.2.2 gg::GgSimpleShader::GgSimpleShader ( ) [inline]

コンストラクタ.

gg.h の 5667 行目に定義があります。

6.13.2.3 gg::GgSimpleShader::GgSimpleShader (const char * *vert*, const char * *frag* = 0, const char * *geom* = 0, GLint *nvarying* = 0, const char ** *varyings* = 0)

コンストラクタ

引数

vert	バーテックスシェーダのソースファイル名.
frag	フラグメントシェーダのソースファイル名 (0 なら不使用).
geom	ジオメトリシェーダのソースファイル名 (0 なら不使用).

nvarying	フィードバックする varying 変数の数 (0 なら不使用).
varyings	フィードバックする varying 変数のリスト.

gg.cpp の 7910 行目に定義があります。

6.13.2.4 gg::GgSimpleShader::GgSimpleShader (const GgSimpleShader & *o*) [inline]

コピーコンストラクタ.

gg.h の 5679 行目に定義があります。

6.13.3 関数詳解

6.13.3.1 void gg::GgSimpleShader::loadLight (const Light & *light*) const [inline]

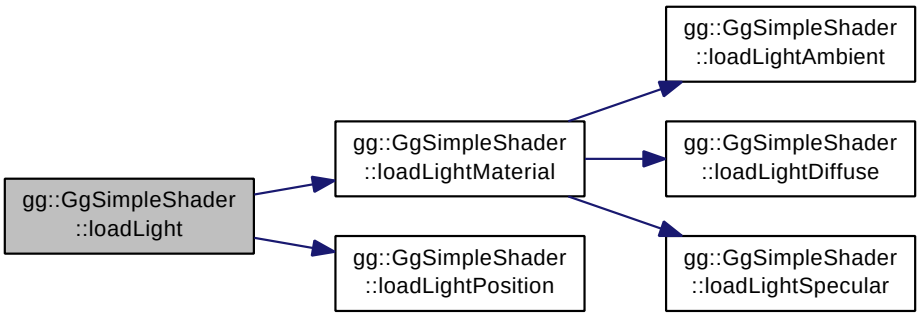
光源の位置と特性を設定する.

引数

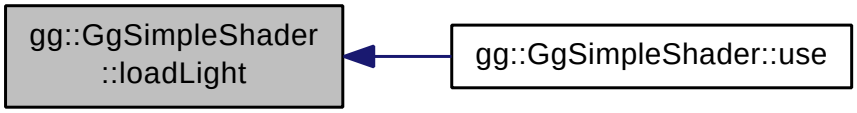
light	光源の特性の gg::GgSimpleShader::Light 構造体

gg.h の 5858 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:



6.13.3.2 void gg::GgSimpleShader::loadLightAmbient (GLfloat *r*, GLfloat *g*, GLfloat *b*, GLfloat *a* = 1.0f) const [inline]

光源の強度の環境光成分を設定する.

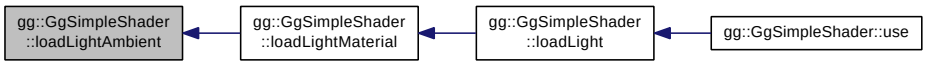
引数

r	光源の強度の環境光成分の赤成分.

g	光源の強度の環境光成分の緑成分.
b	光源の強度の環境光成分の青成分.
a	光源の強度の環境光成分の不透明度. デフォルトは 1.

gg.h の 5775 行目に定義があります。

被呼び出し関係図:



6.13.3.3 void gg::GgSimpleShader::loadLightAmbient (const GLfloat * *amb*) const [inline]

光源の強度の環境光成分を設定する.

引数

amb	光源の強度の環境光成分を格納した GLfloat 型の 4 要素の配列.

gg.h の 5782 行目に定義があります。

6.13.3.4 void gg::GgSimpleShader::loadLightDiffuse (GLfloat *r*, GLfloat *g*, GLfloat *b*, GLfloat *a* = 1.0f) const [inline]

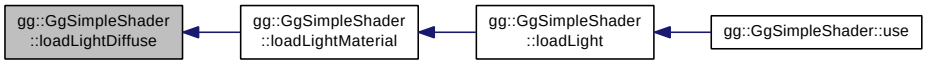
光源の強度の拡散反射光成分を設定する.

引数

r	光源の強度の拡散反射光成分の赤成分.
g	光源の強度の拡散反射光成分の緑成分.
b	光源の強度の拡散反射光成分の青成分.
a	光源の強度の拡散反射光成分の不透明度. デフォルトは 1.

gg.h の 5792 行目に定義があります。

被呼び出し関係図:



6.13.3.5 void gg::GgSimpleShader::loadLightDiffuse (const GLfloat * *diff*) const [inline]

光源の強度の拡散反射光成分を設定する.

引数

diff	光源の強度の拡散反射光成分を格納した GLfloat 型の 4 要素の配列.

gg.h の 5799 行目に定義があります。

6.13.3.6 void gg::GgSimpleShader::loadLightMaterial (const Light & *light*) const [inline]

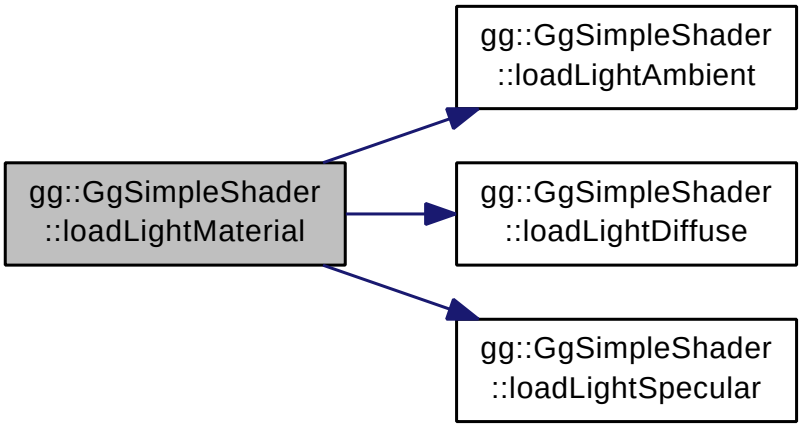
光源の特性を設定する.

引数

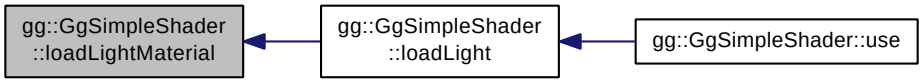
light	光源の特性の gg::GgSimpleShader::Light 構造体

gg.h の 5849 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:



6.13.3.7 void gg::GgSimpleShader::loadLightPosition (GLfloat *x*, GLfloat *y*, GLfloat *z*, GLfloat *w* = `1.0f`) const `[inline]`

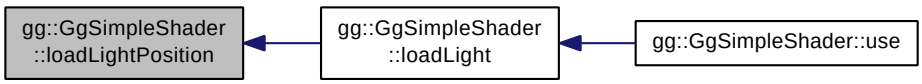
光源の位置を設定する.

引数

x	光源の位置の x 座標.
y	光源の位置の y 座標.
z	光源の位置の z 座標.
w	光源の位置の w 座標. デフォルトは 1.

gg.h の 5826 行目に定義があります。

被呼び出し関係図:



6.13.3.8 void gg::GgSimpleShader::loadLightPosition (const GLfloat * *pos*) const `[inline]`

光源の位置を設定する.

引数

pos	光源の位置の同次座標を格納した GLfloat 型の 4 要素の配列.

gg.h の 5833 行目に定義があります。

6.13.3.9 void gg::GgSimpleShader::loadLightSpecular (GLfloat *r*, GLfloat *g*, GLfloat *b*, GLfloat *a* = `1.0f`) const [inline]

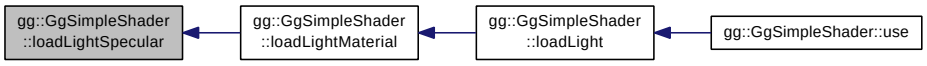
光源の強度の鏡面反射光成分を設定する.

引数

r	光源の強度の鏡面反射光成分の赤成分.
g	光源の強度の鏡面反射光成分の緑成分.
b	光源の強度の鏡面反射光成分の青成分.
a	光源の強度の鏡面反射光成分の不透明度. デフォルトは 1.

gg.h の 5809 行目に定義があります。

被呼び出し関係図:



6.13.3.10 void gg::GgSimpleShader::loadLightSpecular (const GLfloat * *spec*) const [inline]

光源の強度の鏡面反射光成分を設定する.

引数

spec	光源の強度の鏡面反射光成分を格納した GLfloat 型の 4 要素の配列.

gg.h の 5816 行目に定義があります。

6.13.3.11 void gg::GgSimpleShader::loadMaterial (const Material & *material*) const [inline]

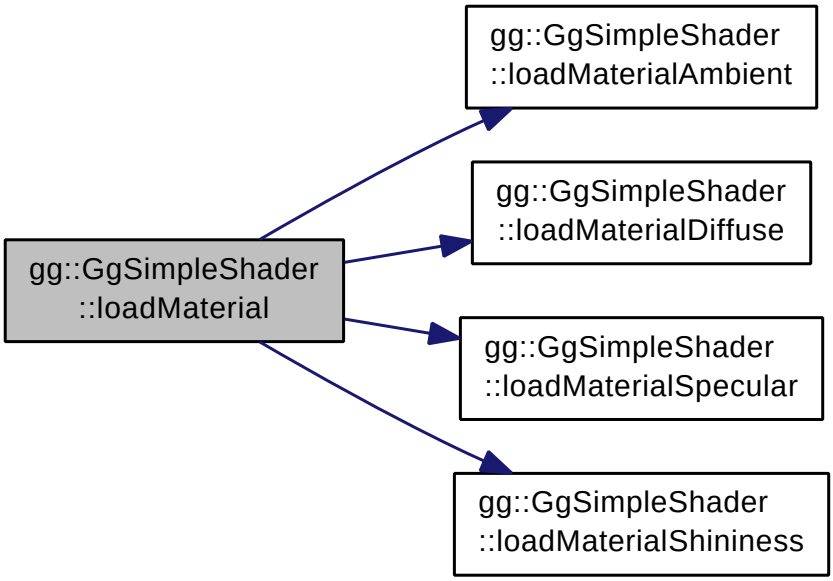
材質の特性を設定する.

引数

material	光源の特性の gg::GgSimpleShader::Material 構造体

gg.h の 5762 行目に定義があります。

呼び出し関係図:



6.13.3.12 void gg::GgSimpleShader::loadMaterialAmbient (GLfloat *r*, GLfloat *g*, GLfloat *b*, GLfloat *a* = `1.0f`) const `[inline]`

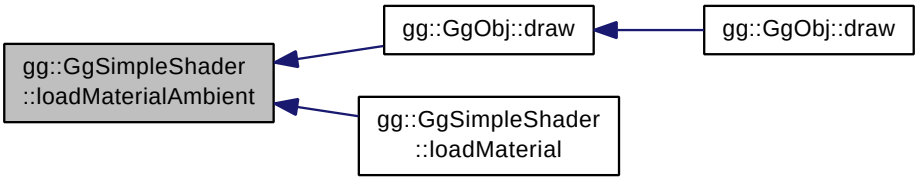
環境光に対する反射係数を設定する.

引数

r	環境光に対する反射係数の赤成分.
g	環境光に対する反射係数の緑成分.
b	環境光に対する反射係数の青成分.
a	環境光に対する反射係数の不透明度. デフォルトは 1.

gg.h の 5698 行目に定義があります。

被呼び出し関係図:



6.13.3.13 void gg::GgSimpleShader::loadMaterialAmbient (const GLfloat * *amb*) const `[inline]`

環境光に対する反射係数を設定する.

引数

amb	環境光に対する反射係数を格納した GLfloat 型の 4 要素の配列.

gg.h の 5705 行目に定義があります。

6.13.3.14 void gg::GgSimpleShader::loadMaterialDiffuse (GLfloat *r*, GLfloat *g*, GLfloat *b*, GLfloat *a* = `1.0f`) const [inline]

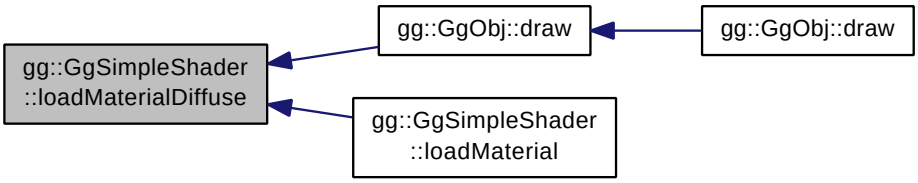
拡散反射係数を設定する.

引数

r	拡散反射係数の赤成分.
g	拡散反射係数の緑成分.
b	拡散反射係数の青成分.
a	拡散反射係数の不透明度. デフォルトは 1.

gg.h の 5715 行目に定義があります。

被呼び出し関係図:



6.13.3.15 void gg::GgSimpleShader::loadMaterialDiffuse (const GLfloat * *diff*) const [inline]

拡散反射係数を設定する.

引数

diff	拡散反射係数を格納した GLfloat 型の 4 要素の配列.

gg.h の 5722 行目に定義があります。

6.13.3.16 void gg::GgSimpleShader::loadMaterialShininess (GLfloat *shi*) const [inline]

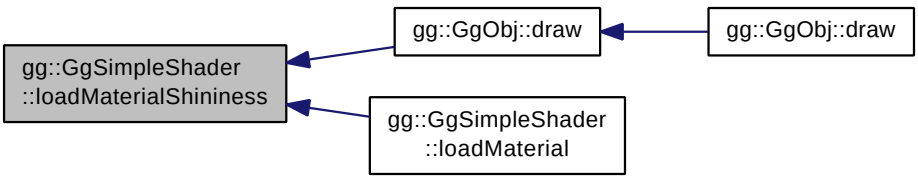
輝き係数を設定する.

引数

shi	輝き係数.

gg.h の 5746 行目に定義があります。

被呼び出し関係図:



6.13.3.17 void gg::GgSimpleShader::loadMaterialSpecular (GLfloat *r*, GLfloat *g*, GLfloat *b*, GLfloat *a* = 1.0f) const [inline]

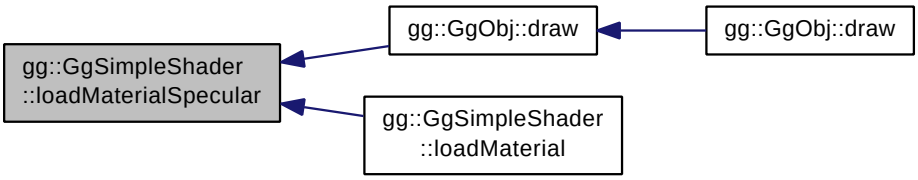
鏡面反射係数を設定する.

引数

r	鏡面反射係数の赤成分.
g	鏡面反射係数の緑成分.
b	鏡面反射係数の青成分.
a	鏡面反射係数の不透明度. デフォルトは 1.

gg.h の 5732 行目に定義があります。

被呼び出し関係図:



6.13.3.18 void gg::GgSimpleShader::loadMaterialSpecular (const GLfloat * *spec*) const [inline]

鏡面反射係数を設定する.

引数

spec	鏡面反射係数を格納した GLfloat 型の 4 要素の配列.

gg.h の 5739 行目に定義があります。

6.13.3.19 virtual void gg::GgSimpleShader::loadMatrix (const GgMatrix & *mp*, const GgMatrix & *mv*, const GgMatrix & *mn*) const [inline], [virtual]

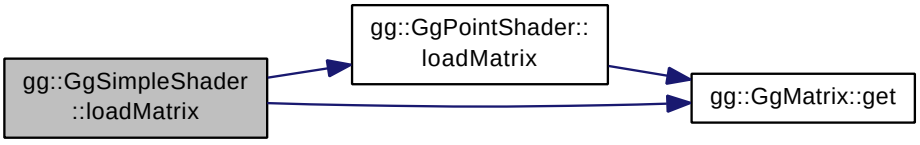
変換行列を設定する.

引数

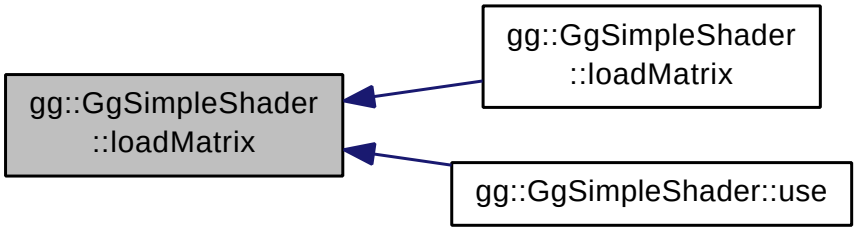
mp	GgMatrix 型の投影変換行列.
mv	GgMatrix 型のモデルビュー変換行列.
mn	GgMatrix 型のモデルビュー変換行列の法線変換行列.

gg.h の 5868 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:



6.13.3.20 virtual void gg::GgSimpleShader::loadMatrix (const GLfloat * *mp*, const GLfloat * *mv*, const GLfloat * *mn*) const [inline], [virtual]

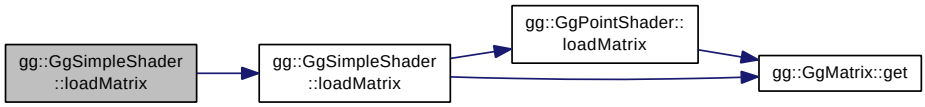
変換行列を設定する.

引数

mp	GLfloat 型の 16 要素の配列に格納された投影変換行列.
mv	GLfloat 型の 16 要素の配列に格納されたモデルビュー変換行列.
mv	GLfloat 型の 16 要素の配列に格納されたモデルビュー変換行列の法線変換行列.

gg.h の 5878 行目に定義があります。

呼び出し関係図:



6.13.3.21 virtual void gg::GgSimpleShader::loadMatrix (const GgMatrix & *mp*, const GgMatrix & *mv*) const [inline], [virtual]

変換行列を設定する.

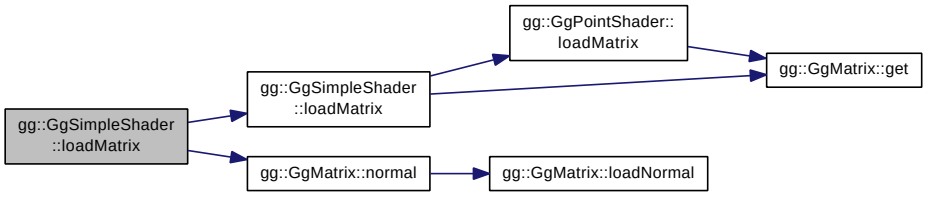
引数

mp	GgMatrix 型の投影変換行列.
mv	GgMatrix 型のモデルビュー変換行列.

gg::GgPointShaderを再実装しています。

gg.h の 5886 行目に定義があります。

呼び出し関係図:



6.13.3.22 virtual void gg::GgSimpleShader::loadMatrix (const GLfloat * *mp*, const GLfloat * *mv*) const [inline], [virtual]

変換行列を設定する.

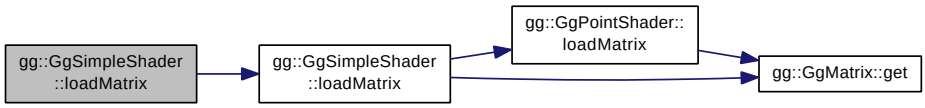
引数

mp	GLfloat 型の 16 要素の配列に格納された投影変換行列.
mv	GLfloat 型の 16 要素の配列に格納されたモデルビュー変換行列.

gg::GgPointShaderを再実装しています。

gg.h の 5894 行目に定義があります。

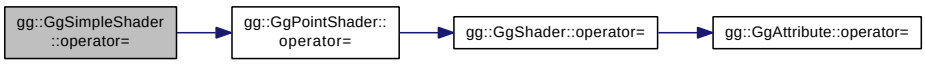
呼び出し関係図:



6.13.3.23 GgSimpleShader& gg::GgSimpleShader::operator= (const GgSimpleShader & *o*) [inline]

gg.h の 5683 行目に定義があります。

呼び出し関係図:

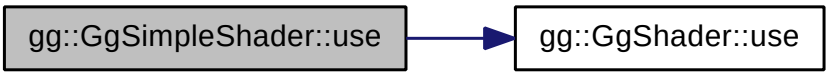


6.13.3.24 void gg::GgSimpleShader::use ( ) const [inline]

シェーダプログラムの使用を開始する.

gg.h の 5900 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:

6.13.3.25 void gg::GgSimpleShader::use (const Light & *light*, const GgMatrix & *mp*, const GgMatrix & *mv*, const GgMatrix & *mn*) const [inline]

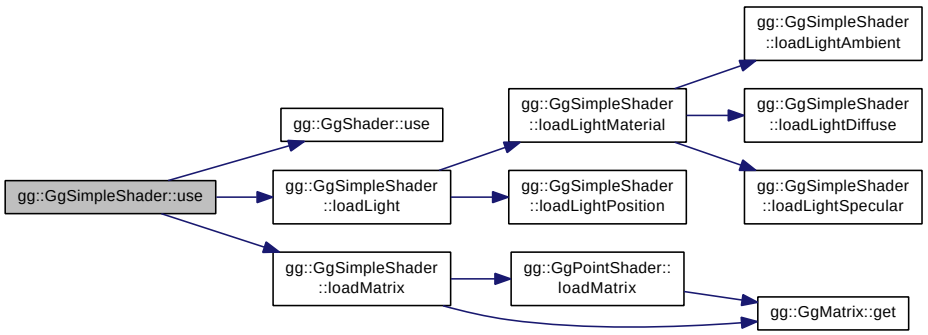
光源と変換行列を指定してシェーダプログラムの使用を開始する.

引数

light	光源の特性の gg::GgSimpleShader::Light 構造体
mp	GgMatrix 型の投影変換行列.
mv	GgMatrix 型のモデルビュー変換行列.
mn	GgMatrix 型のモデルビュー変換行列の法線変換行列.

gg.h の 5911 行目に定義があります。

呼び出し関係図:



6.13.3.26 void gg::GgSimpleShader::use (const Light & *light*, const GLfloat * *mp*, const GLfloat * *mv*, const GLfloat * *mn*) const [inline]

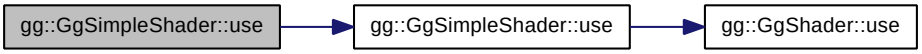
光源と変換行列を指定してシェーダプログラムの使用を開始する.

引数

light	光源の特性の gg::GgSimpleShader::Light 構造体
mp	GLfloat 型の 16 要素の配列に格納された投影変換行列.
mv	GLfloat 型の 16 要素の配列に格納されたモデルビュー変換行列.
mv	GLfloat 型の 16 要素の配列に格納されたモデルビュー変換行列の法線変換行列.

gg.h の 5928 行目に定義があります。

呼び出し関係図:



6.13.3.27 void gg::GgSimpleShader::use (const Light & *light*, const GgMatrix & *mp*, const GgMatrix & *mv*) const [inline]

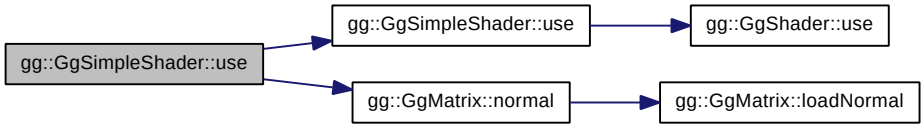
光源と変換行列を指定してシェーダプログラムの使用を開始する.

引数

light	光源の特性の gg::GgSimpleShader::Light 構造体
mp	GgMatrix 型の投影変換行列.
mv	GgMatrix 型のモデルビュー変換行列.

gg.h の 5937 行目に定義があります。

呼び出し関係図:



6.13.3.28 void gg::GgSimpleShader::use (const Light & *light*, const GLfloat * *mp*, const GLfloat * *mv*) const [inline]

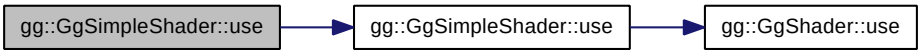
光源と変換行列を指定してシェーダプログラムの使用を開始する.

引数

light	光源の特性の gg::GgSimpleShader::Light 構造体
mp	GLfloat 型の 16 要素の配列に格納された投影変換行列.
mv	GLfloat 型の 16 要素の配列に格納されたモデルビュー変換行列.

gg.h の 5946 行目に定義があります。

呼び出し関係図:



このクラス詳解は次のファイルから抽出されました:

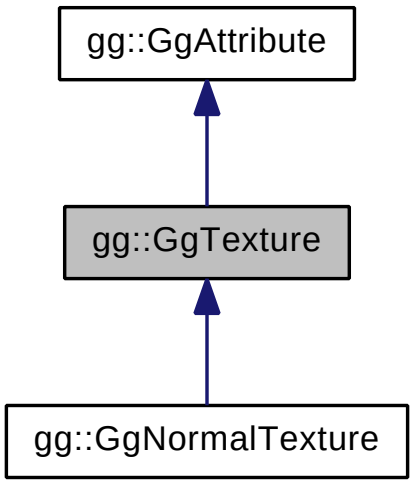
- gg.h
- gg.cpp

6.14 gg::GgTexture クラス

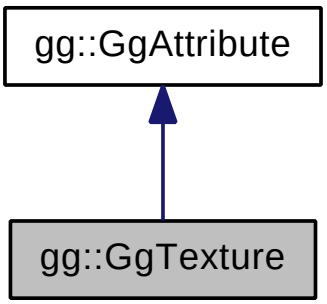
テクスチャ.

`#include <gg.h>`

gg::GgTexture の継承関係図



gg::GgTexture 連携図



公開メンバ関数

- virtual ~GgTexture ()
  デストラクタ.
- GgTexture ()
  コンストラクタ.
- GgTexture (GLuint tex)
  コンストラクタ.
- GgTexture (GLsizei width, GLsizei height, GLenum internal=GL_RGBA, GLenum format=GL_RGBA, const GLvoid *image=NULL)
  コンストラクタ.
- GgTexture (const char *name, GLenum internal=GL_RGBA)
  コンストラクタ.
- GgTexture (const GgTexture &o)
- GgTexture & operator= (const GgTexture &o)
- void use () const
  テクスチャの使用開始 *(**このテクスチャを使用する際に呼び出す**)*.
- void unuse () const
  テクスチャの使用終了 *(**このテクスチャを使用しなくなったら呼び出す**)*.
- GLuint get () const
  使用しているテクスチャのテクスチャ名を得る.

その他の継承メンバ

6.14.1 詳解

テクスチャ.

カラー画像を読み込んでテクスチャマップを作成する.

gg.h の 4789 行目に定義があります。

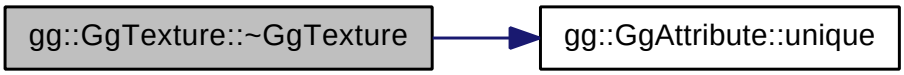
6.14.2 構築子と解体子

6.14.2.1 virtual gg::GgTexture::~GgTexture ( ) [inline], [virtual]

デストラクタ.

gg.h の 4798 行目に定義があります。

呼び出し関係図:



6.14.2.2 gg::GgTexture::GgTexture ( ) [inline]

コンストラクタ.

gg.h の 4809 行目に定義があります。

6.14.2.3 gg::GgTexture::GgTexture (GLuint *tex*) [inline]

コンストラクタ.

gg.h の 4815 行目に定義があります。

6.14.2.4 gg::GgTexture::GgTexture (GLsizei *width*, GLsizei *height*, GLenum *internal* = GL_RGBA, GLenum *format* = GL_RGBA, const GLvoid * *image* = NULL) [inline]

コンストラクタ.

引数

width	テクスチャの幅.
height	テクスチャの高さ.
internal	テクスチャの内部フォーマット.
format	読み込むテクスチャのフォーマット.
image	テクスチャとして用いる画像データ (0 ならデータを読み込まずテクスチャメモリの確保だけを行う)

gg.h の 4824 行目に定義があります。

6.14.2.5 gg::GgTexture::GgTexture (const char * *name*, GLenum *internal* = GL_RGBA) [inline]

コンストラクタ.

引数

name	テクスチャメモリに読み込む TGA フォーマットの画像ファイル名.
internal	テクスチャの内部フォーマット.

gg.h の 4831 行目に定義があります。

6.14.2.6 gg::GgTexture::GgTexture (const GgTexture & *o*) [inline]

gg.h の 4835 行目に定義があります。

6.14.3 関数詳解

6.14.3.1 GLuint gg::GgTexture::get ( ) const [inline]

使用しているテクスチャのテクスチャ名を得る.

戻り値

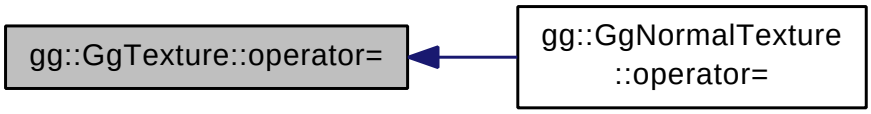
テクスチャ名.

gg.h の 4863 行目に定義があります。

6.14.3.2 GgTexture& gg::GgTexture::operator= (const GgTexture & *o*) [inline]

gg.h の 4839 行目に定義があります。
呼び出し関係図:

被呼び出し関係図:



6.14.3.3 void gg::GgTexture::unuse ( ) const [inline]

テクスチャの使用終了 (このテクスチャを使用しなくなったら呼び出す).
gg.h の 4856 行目に定義があります。

6.14.3.4 void gg::GgTexture::use ( ) const [inline]

テクスチャの使用開始 (このテクスチャを使用する際に呼び出す).
gg.h の 4850 行目に定義があります。
このクラス詳解は次のファイルから抽出されました:

- gg.h

6.15 gg::GgTrackball クラス

簡易トラックボール処理.

`#include <gg.h>`

公開メンバ関数

- virtual ~GgTrackball ()
  デストラクタ.
- GgTrackball ()
  コンストラクタ.
- void region (float w, float h)

トラックボール処理するマウスの移動範囲を指定する.

- void region (int w, int h)
  トラックボール処理するマウスの移動範囲を指定する.
- void start (float x, float y)
  トラックボール処理を開始する.
- void motion (float x, float y)
  回転の変換行列を計算する.
- void rotate (const GgQuaternion &q)
  トラックボールの回転角を修正する.
- void stop (float x, float y)
  トラックボール処理を停止する.
- void reset ()
  トラックボールをリセットする
- const GLfloat * get () const
  現在の回転の変換行列を取り出す.
- const GgMatrix & getMatrix () const
  現在の回転の変換行列を取り出す.
- const GgQuaternion & getQuaternion () const
  現在の回転の四元数を取り出す.

6.15.1 詳解

簡易トラックボール処理.

gg.h の 4615 行目に定義があります。

6.15.2 構築子と解体子

6.15.2.1 virtual gg::GgTrackball::~GgTrackball ( ) [inline], [virtual]

デストラクタ.

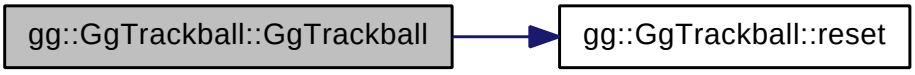
gg.h の 4627 行目に定義があります。

6.15.2.2 gg::GgTrackball::GgTrackball ( ) [inline]

コンストラクタ.

gg.h の 4630 行目に定義があります。

呼び出し関係図:



6.15.3 関数詳解

6.15.3.1 const GLfloat* gg::GgTrackball::get ( ) const [inline]

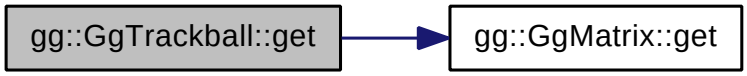
現在の回転の変換行列を取り出す.

戻り値

回転の変換を表す GLfloat 型の 16 要素の配列.

gg.h の 4677 行目に定義があります。

呼び出し関係図:



6.15.3.2 const GgMatrix& gg::GgTrackball::getMatrix ( ) const [inline]

現在の回転の変換行列を取り出す.

戻り値

回転の変換を表す GgMatrix 型の変換行列.

gg.h の 4684 行目に定義があります。

6.15.3.3 const GgQuaternion& gg::GgTrackball::getQuaternion ( ) const [inline]

現在の回転の四元数を取り出す.

戻り値

回転の変換を表す Quaternion 型の四元数.

gg.h の 4691 行目に定義があります。

6.15.3.4 void gg::GgTrackball::motion (float *x*, float *y*)

回転の変換行列を計算する.

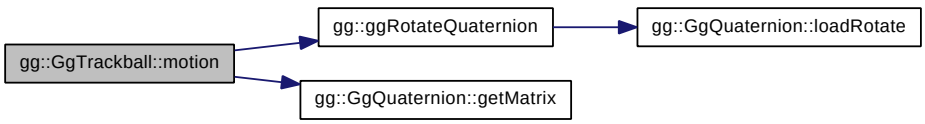
マウスのドラッグ中に呼び出す.

引数

x	現在のマウスの x 座標.
y	現在のマウスの y 座標.

gg.cpp の 7462 行目に定義があります。

呼び出し関係図:



6.15.3.5 void gg::GgTrackball::region (float *w*, float *h*)

トラックボール処理するマウスの移動範囲を指定する.

ウィンドウのリサイズ時に呼び出す.

引数

w	領域の幅.
h	領域の高さ.

gg.cpp の 7433 行目に定義があります。

被呼び出し関係図:

6.15.3.6 void gg::GgTrackball::region (int *w*, int *h*) [inline]

トラックボール処理するマウスの移動範囲を指定する.

ウィンドウのリサイズ時に呼び出す.

引数

w	領域の幅.
h	領域の高さ.

gg.h の 4645 行目に定義があります。

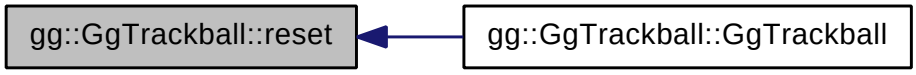
呼び出し関係図:

6.15.3.7 void gg::GgTrackball::reset ( )

トラックボールをリセットする

gg.cpp の 7415 行目に定義があります。

被呼び出し関係図:



6.15.3.8 void gg::GgTrackball::rotate (const GgQuaternion & *q*)

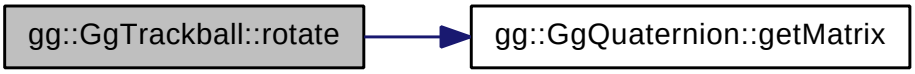
トラックボールの回転角を修正する.

引数

q	修正分の回転角の四元数.

gg.cpp の 7490 行目に定義があります。

呼び出し関係図:



6.15.3.9 void gg::GgTrackball::start (float *x*, float *y*)

トラックボール処理を開始する.

マウスのドラッグ開始時 (マウスボタンを押したとき) に呼び出す.

引数

x	現在のマウスの x 座標.
y	現在のマウスの y 座標.

gg.cpp の 7446 行目に定義があります。

6.15.3.10 void gg::GgTrackball::stop (float *x*, float *y*)

トラックボール処理を停止する.

マウスのドラッグ終了時 (マウスボタンを離したとき) に呼び出す.

引数

x	現在のマウスの x 座標.
y	現在のマウスの y 座標.

gg.cpp の 7511 行目に定義があります。

このクラス詳解は次のファイルから抽出されました:

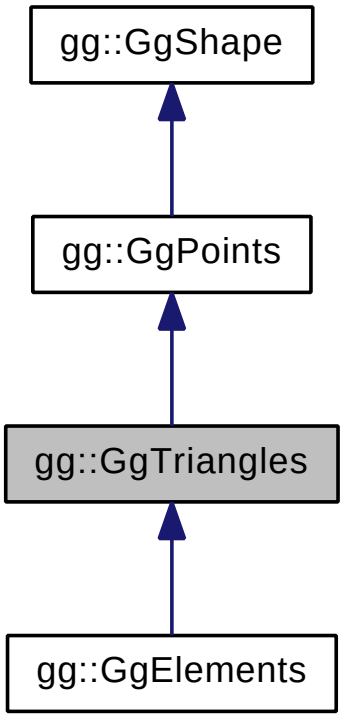
- gg.h
- gg.cpp

6.16 gg::GgTriangles クラス

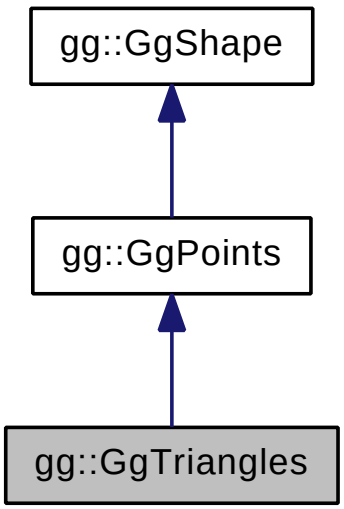
三角形で表した形状データ (Arrays 形式).

`#include <gg.h>`

gg::GgTriangles の継承関係図



gg::GgTriangles 連携図



公開メンバ関数

- virtual ~GgTriangles ()
  デストラクタ.
- GgTriangles (GLenum mode=GL_TRIANGLES)
  コンストラクタ
- GgTriangles (GLuint nv, const GLfloat(∗pos)[3], const GLfloat(∗norm)[3], GLenum mode=GL_TRIANGLES, GLenum usage=GL_STATIC_DRAW)
  コンストラクタ.

- GgTriangles (const GgTriangles &o)
  コピーコンストラクタ.
- GgTriangles & operator= (const GgTriangles &o)
- void send (GLuint nv, const GLfloat(∗pos)[3], const GLfloat(∗norm)[3], GLuint offset=0)
  既存のバッファオブジェクトに頂点の位置データと法線データを転送する.
- void load (GLuint nv, const GLfloat(∗pos)[3], const GLfloat(∗norm)[3], GLenum usage=GL_STATIC_DRAW)
  バッファオブジェクトを確保して頂点の位置データと法線データを格納する.
- GLuint nbuf () const
  頂点の位置データを格納した頂点バッファオブジェクト名を取り出す.
- GLuint nnum () const
  データの数を取り出す.

6.16.1 詳解

三角形で表した形状データ (Arrays 形式).

gg.h の 5198 行目に定義があります。

6.16.2 構築子と解体子

6.16.2.1 virtual gg::GgTriangles::∼GgTriangles ( ) [inline], [virtual]

デストラクタ.

gg.h の 5221 行目に定義があります。

6.16.2.2 gg::GgTriangles::GgTriangles (GLenum *mode* = GL_TRIANGLES) [inline]

コンストラクタ

引数

mode	描画する基本図形の種類.

gg.h の 5225 行目に定義があります。

6.16.2.3 gg::GgTriangles::GgTriangles (GLuint *nv*, const GLfloat(∗) *pos[3]*, const GLfloat(∗) *norm[3]*, GLenum *mode* = GL_TRIANGLES, GLenum *usage* = GL_STATIC_DRAW) [inline]

コンストラクタ.

引数

nv	頂点数.
pos	この図形の頂点の位置のデータの配列 (NULL ならデータを転送しない).
norm	この図形の頂点の法線のデータの配列 (NULL ならデータを転送しない).
mode	描画する基本図形の種類.
usage	バッファオブジェクトの使い方.

gg.h の 5234 行目に定義があります。

6.16.2.4 gg::GgTriangles::GgTriangles (const GgTriangles & *o*) [inline]

コピーコンストラクタ.

gg.h の 5242 行目に定義があります。

6.16.3 関数詳解

6.16.3.1 void gg::GgTriangles::load (GLuint *nv*, const GLfloat(∗) *pos[3]*, const GLfloat(∗) *norm[3]*, GLenum *usage* = GL_STATIC_DRAW) [inline]

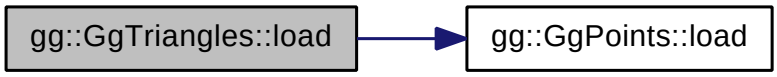
バッファオブジェクトを確保して頂点の位置データと法線データを格納する.

引数

nv	頂点のデータの数 (頂点数).
pos	頂点の位置データが格納されてている領域の先頭のポインタ.
norm	頂点の法線データが格納されてている領域の先頭のポインタ.
usage	バッファオブジェクトの使い方.

gg.h の 5272 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:

6.16.3.2 GLuint gg::GgTriangles::nbuf ( ) const [inline]

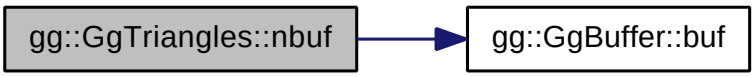
頂点の位置データを格納した頂点バッファオブジェクト名を取り出す.

戻り値

この図形の頂点の法線のデータを格納した頂点バッファオブジェクト名.

gg.h の 5280 行目に定義があります。

呼び出し関係図:



6.16.3.3 GLuint gg::GgTriangles::nnum ( ) const [inline]

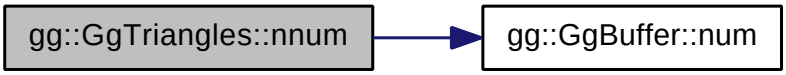
データの数を取り出す.

戻り値

この図形の頂点の法線データの数 (頂点数).

gg.h の 5287 行目に定義があります。

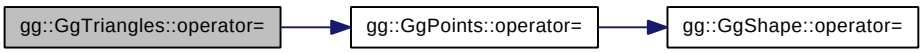
呼び出し関係図:



6.16.3.4 GgTriangles& gg::GgTriangles::operator= (const GgTriangles & *o*) [inline]

gg.h の 5246 行目に定義があります。

呼び出し関係図:



被呼び出し関係図:

6.16.3.5 void gg::GgTriangles::send (GLuint *nv*, const GLfloat(∗) *pos[3]*, const GLfloat(∗) *norm[3]*, GLuint *offset* = 0) [inline]

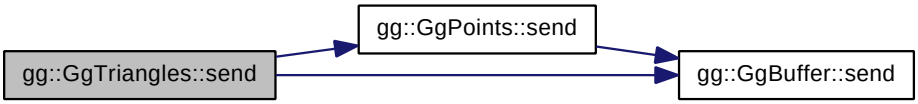
既存のバッファオブジェクトに頂点の位置データと法線データを転送する.

引数

nv	転送する頂点の位置データの数 (0 ならバッファ全体).
pos	転送元の頂点の位置データが格納されてている領域の先頭のポインタ.
norm	転送元の頂点の法線データが格納されてている領域の先頭のポインタ.
offset	転送先のバッファオブジェクトの先頭の要素番号.

gg.h の 5261 行目に定義があります。

呼び出し関係図:



このクラス詳解は次のファイルから抽出されました:

- gg.h

6.17 gg::GgSimpleShader::Light 構造体

光源の特性

```
#include <gg.h>
```

公開変数類

- GLfloat ambient [4]
- GLfloat diffuse [4]
- GLfloat specular [4]
- GLfloat position [4]

6.17.1 詳解

光源の特性

gg.h の 5839 行目に定義があります。

6.17.2 メンバ詳解

6.17.2.1 GLfloat gg::GgSimpleShader::Light::ambient[4]

gg.h の 5841 行目に定義があります。

6.17.2.2 GLfloat gg::GgSimpleShader::Light::diffuse[4]

gg.h の 5842 行目に定義があります。

6.17.2.3 GLfloat gg::GgSimpleShader::Light::position[4]

gg.h の 5844 行目に定義があります。

6.17.2.4 GLfloat gg::GgSimpleShader::Light::specular[4]

gg.h の 5843 行目に定義があります。

この構造体詳解は次のファイルから抽出されました:

- gg.h

6.18 gg::GgSimpleShader::Material 構造体

材質の特性

```
#include <gg.h>
```

公開変数類

- GLfloat ambient [4]
- GLfloat diffuse [4]
- GLfloat specular [4]
- GLfloat shininess

6.18.1 詳解

材質の特性

gg.h の 5752 行目に定義があります。

6.18.2 メンバ詳解

6.18.2.1 GLfloat gg::GgSimpleShader::Material::ambient[4]

gg.h の 5754 行目に定義があります。

6.18.2.2 GLfloat gg::GgSimpleShader::Material::diffuse[4]

gg.h の 5755 行目に定義があります。

6.18.2.3 GLfloat gg::GgSimpleShader::Material::shininess

gg.h の 5757 行目に定義があります。

6.18.2.4 GLfloat gg::GgSimpleShader::Material::specular[4]

gg.h の 5756 行目に定義があります。
この構造体詳解は次のファイルから抽出されました:

- gg.h

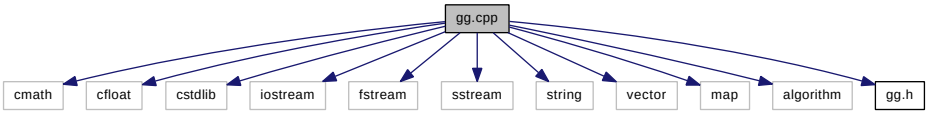
Chapter 7

ファイル詳解

7.1 gg.cpp ファイル

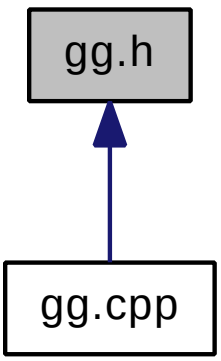
```
#include <cmath>
#include <cfloat>
#include <cstdlib>
#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include <vector>
#include <map>
#include <algorithm>
#include "gg.h"
```

gg.cpp の依存先関係図:



7.2 gg.h ファイル

被依存関係図:



クラス

- class gg::GgMatrix
  変換行列.
- class gg::GgQuaternion
  四元数.
- class gg::GgTrackball
  簡易トラックボール処理.
- class gg::GgCounter
  参照カウンタ.
- class gg::GgAttribute
  属性データ.
- class gg::GgTexture
  テクスチャ.
- class gg::GgNormalTexture
  法線マップ.
- class gg::GgBuffer< T >
  バッファオブジェクト.
- class gg::GgShape
  形状データの基底クラス.
- class gg::GgPoints
  ポイント.
- class gg::GgTriangles
  三角形で表した形状データ *(Arrays* **形式***)*.
- class gg::GgElements
  三角形で表した形状データ *(Elements* **形式***)*.
- class gg::GgShader
  シェーダの基底クラス.
- class gg::GgPointShader
  ポイントのシェーダ.
- class gg::GgSimpleShader
  三角形に単純な陰影付けを行うシェーダ.
- struct gg::GgSimpleShader::Material
  材質の特性
- struct gg::GgSimpleShader::Light
  光源の特性
- class gg::GgObj
  Wavefront OBJ **形式のファイル**.

名前空間

- gg

関数

- void gg::ggInit ()
  ゲームグラフィックス特論の都合にもとづく初期化を行う.
- void gg::ggError (const char *name=NULL, unsigned int line=0)
  OpenGL **のエラーをチェックする**.
- void gg::ggFBOError (const char *name=NULL, unsigned int line=0)

FBO のエラーをチェックする.

- bool gg::ggSaveTga (GLsizei sx, GLsizei sy, unsigned int depth, const void *buffer, const char *name)

  配列の内容を *TGA* ファイルに保存する.
- bool gg::ggSaveColor (const char *name)

  カラーバッファの内容を *TGA* ファイルに保存する.
- bool gg::ggSaveDepth (const char *name)

  デプスバッファの内容を *TGA* ファイルに保存する.
- GLubyte * gg::ggLoadTga (const char *name, GLsizei *width, GLsizei *height, GLenum *format)

  TGA ファイル *(8/16/24/32bit)* をメモリに読み込む.
- GLuint gg::ggLoadTexture (GLsizei width, GLsizei height, GLenum internal, GLenum format=GL_RGBA, const GLvoid *image=NULL)

  テクスチャメモリを確保して画像データをテクスチャとして読み込む.
- GLuint gg::ggLoadImage (const char *name, GLenum internal=0)

  TGA 画像ファイルをテクスチャとして読み込む.
- GLuint gg::ggLoadHeight (const char *name, float nz, GLenum internal=GL_RGBA)

  高さマップ用の *TGA* 画像ファイルの読み込んで法線マップを作成する.
- bool gg::ggLoadObj (const char *name, GLuint &nv, GLfloat(*&pos)[3], GLfloat(*&norm)[3], GLuint &nf, GLuint(*&face)[3], bool normalize=false)

  三角形分割された *OBJ* ファイルを読み込む *(Elements* 形式*)*.
- bool gg::ggLoadObj (const char *name, GLuint &ng, GLuint(*&group)[2], GLfloat(*&amb)[4], GLfloat(*&diff)[4], GLfloat(*&spec)[4], GLfloat *&shi, GLuint &nv, GLfloat(*&pos)[3], GLfloat(*&norm)[3], bool normalize=false)

  三角形分割された *OBJ* ファイルと *MTL* ファイルを読み込む *(Arrays* 形式*)*
- GLuint gg::ggCreateShader (const char *vsrc, const char *fsrc=NULL, const char *gsrc=NULL, GLint nvarying=0, const char *const varyings[]=NULL, const char *vtext="vertex shader", const char *ftext="fragment shader", const char *gtext="geometry shader")

  シェーダのソースプログラムの文字列を読み込んでプログラムオブジェクトを作成する.
- GLuint gg::ggLoadShader (const char *vert, const char *frag=NULL, const char *geom=NULL, GLint nvarying=0, const char *const varyings[]=NULL)

  シェーダのソースファイルを読み込んでプログラムオブジェクトを作成する.
- GLfloat gg::ggLength3 (const GLfloat *a)

  3 要素の長さ.
- GLfloat gg::ggLength4 (const GLfloat *a)

  4 要素の長さ.
- GLfloat gg::ggDot3 (const GLfloat *a, const GLfloat *b)

  3 要素の内積.
- void gg::ggCross (GLfloat *c, const GLfloat *a, const GLfloat *b)

  3 要素の外積.
- GLfloat gg::ggDot4 (const GLfloat *a, const GLfloat *b)

  4 要素の内積
- GgMatrix gg::ggIdentity ()

  単位行列を返す.
- GgMatrix gg::ggTranslate (GLfloat x, GLfloat y, GLfloat z, GLfloat w=1.0f)

  平行移動の変換行列を返す.
- GgMatrix gg::ggTranslate (const GLfloat *t)

  平行移動の変換行列を返す.
- GgMatrix gg::ggScale (GLfloat x, GLfloat y, GLfloat z, GLfloat w=1.0f)

  拡大縮小の変換行列を返す.
- GgMatrix gg::ggScale (const GLfloat *s)

  拡大縮小の変換行列を返す.
- GgMatrix gg::ggRotateX (GLfloat a)

  x 軸中心の回転の変換行列を返す.

- GgMatrix gg::ggRotateY (GLfloat a)
  y **軸中心の回転の変換行列を返す**.
- GgMatrix gg::ggRotateZ (GLfloat a)
  z **軸中心の回転の変換行列を返す**.
- GgMatrix gg::ggRotate (GLfloat x, GLfloat y, GLfloat z, GLfloat a)
  (x, y, z) **方向のベクトルを軸とする回転の変換行列を乗じた結果を返す**.
- GgMatrix gg::ggRotate (const GLfloat *r, GLfloat a)
  r **方向のベクトルを軸とする回転の変換行列を乗じた結果を返す**.
- GgMatrix gg::ggRotate (const GLfloat *r)
  r **方向のベクトルを軸とする回転の変換行列を乗じた結果を返す**.
- GgMatrix gg::ggLookat (GLfloat ex, GLfloat ey, GLfloat ez, GLfloat tx, GLfloat ty, GLfloat tz, GLfloat ux, GLfloat uy, GLfloat uz)
  ビュー変換行列を返す.
- GgMatrix gg::ggLookat (const GLfloat *e, const GLfloat *t, const GLfloat *u)
  ビュー変換行列を返す.
- GgMatrix gg::ggOrthogonal (GLfloat left, GLfloat right, GLfloat bottom, GLfloat top, GLfloat zNear, GLfloat zFar)
  直交投影変換行列を返す.
- GgMatrix gg::ggFrustum (GLfloat left, GLfloat right, GLfloat bottom, GLfloat top, GLfloat zNear, GLfloat zFar)
  透視透視投影変換行列を返す.
- GgMatrix gg::ggPerspective (GLfloat fovy, GLfloat aspect, GLfloat zNear, GLfloat zFar)
  画角を指定して透視投影変換行列を返す.
- GgMatrix gg::ggTranspose (const GgMatrix &m)
  転置行列を返す.
- GgMatrix gg::ggInvert (const GgMatrix &m)
  逆行列を返す.
- GgMatrix gg::ggNormal (const GgMatrix &m)
  法線変換行列を返す.
- GgQuaternion gg::ggQuaternion (GLfloat x, GLfloat y, GLfloat z, GLfloat w)
  四元数を返す
- GgQuaternion gg::ggQuaternion (const GLfloat *a)
  四元数を返す
- GgQuaternion gg::ggIdentityQuaternion ()
  単位四元数を返す
- GgQuaternion gg::ggMatrixQuaternion (const GLfloat *a)
  回転の変換行列 *m* **を表す四元数を返す**.
- GgQuaternion gg::ggMatrixQuaternion (const GgMatrix &m)
  回転の変換行列 *m* **を表す四元数を返す**.
- GgMatrix gg::ggQuaternionMatrix (const GgQuaternion &q)
  四元数 *q* **の回転の変換行列を返す**.
- GgMatrix gg::ggQuaternionTransposeMatrix (const GgQuaternion &q)
  四元数 *q* **の回転の転置した変換行列を返す**.
- GgQuaternion gg::ggRotateQuaternion (GLfloat x, GLfloat y, GLfloat z, GLfloat a)
  (x, y, z) **を軸として角度** *a* **回転する四元数を返す**.
- GgQuaternion gg::ggRotateQuaternion (const GLfloat *v, GLfloat a)
  (v[0], v[1], v[2]) **を軸として角度** *a* **回転する四元数を返す**.
- GgQuaternion gg::ggRotateQuaternion (const GLfloat *v)
  (v[0], v[1], v[2]) **を軸として角度** *v[3]* **回転する四元数を返す**.
- GgQuaternion gg::ggEulerQuaternion (GLfloat heading, GLfloat pitch, GLfloat roll)
  オイラー角 *(heading, pitch, roll)* **で与えられた回転を表す四元数を返す**.
- GgQuaternion gg::ggEulerQuaternion (const GLfloat *e)

オイラー角 *(e[0], e[1], e[2])* **で与えられた回転を表す四元数を返す**.

- GgQuaternion gg::ggSlerp (const GLfloat ∗a, const GLfloat ∗b, GLfloat t)

  二つの四元数の球面線形補間の結果を返す.
- GgQuaternion gg::ggSlerp (const GgQuaternion &q, const GgQuaternion &r, GLfloat t)

  二つの四元数の球面線形補間の結果を返す.
- GgQuaternion gg::ggSlerp (const GgQuaternion &q, const GLfloat ∗a, GLfloat t)

  二つの四元数の球面線形補間の結果を返す.
- GgQuaternion gg::ggSlerp (const GLfloat ∗a, const GgQuaternion &q, GLfloat t)

  二つの四元数の球面線形補間の結果を返す.
- GLfloat gg::ggNorm (const GgQuaternion &q)

  四元数のノルムを返す.
- GgQuaternion gg::ggNormalize (const GgQuaternion &q)

  正規化した四元数を返す.
- GgQuaternion gg::ggConjugate (const GgQuaternion &q)

  共役四元数を返す.
- GgQuaternion gg::ggInvert (const GgQuaternion &q)

  四元数の逆元を求める.
- GgPoints ∗ gg::ggPointsCube (GLuint nv, GLfloat length=1.0f, GLfloat cx=0.0f, GLfloat cy=0.0f, GLfloat cz=0.0f)

  点群を立方体状に生成する.
- GgPoints ∗ gg::ggPointsSphere (GLuint nv, GLfloat radius=0.5f, GLfloat cx=0.0f, GLfloat cy=0.0f, GLfloat cz=0.0f)

  点群を球状に生成する.
- GgTriangles ∗ gg::ggRectangle (GLfloat width=1.0f, GLfloat height=1.0f)

  矩形状に *2* **枚の三角形を生成する**.
- GgTriangles ∗ gg::ggEllipse (GLfloat width=1.0f, GLfloat height=1.0f, GLuint slices=16)

  楕円状に三角形を生成する.
- GgTriangles ∗ gg::ggArraysObj (const char ∗name, bool normalize=false)

  Wavefront OBJ **ファイルを読み込む** *(Arrays* **形式***)*
- GgElements ∗ gg::ggElementsObj (const char ∗name, bool normalize=false)

  Wavefront OBJ **ファイル を読み込む** *(Elements* **形式***).*
- GgElements ∗ gg::ggElementsMesh (int slices, int stacks, const GLfloat(∗pos)[3], const GLfloat(∗norm)[3]=NULL)

  メッシュ形状を作成する *(Elements* **形式***).*
- GgElements ∗ gg::ggElementsSphere (GLfloat radius=1.0f, int slices=16, int stacks=8)

  球状に三角形データを生成する *(Elements* **形式***).*