```
#ifndef __glu_h__
#define __glu_h__

#if defined(USE_MGL_NAMESPACE)
#include "glu_mangle.h"
#endif

#include <GL/gl.h>

#ifndef GLAPIENTRY
#if defined(_MSC_VER) || defined(__MINGW32__)
#define GLAPIENTRY __stdcall
#else
#define GLAPIENTRY
#endif
#endif

#ifndef GLAPIENTRYP
```

```
#define GLAPIENTRYP GLAPIENTRY *
#endif

#if (defined(_MSC_VER) || defined(__MINGW32__)) && defined(BUILD_GLU32)
# undef GLAPI
# define GLAPI __declspec(dllexport)
#elif (defined(_MSC_VER) || defined(__MINGW32__)) && defined(_DLL)
/* tag specifying we're building for DLL runtime support */
# undef GLAPI
# define GLAPI __declspec(dllimport)
#elif !defined(GLAPI)
/* for use with static link lib build of Win32 edition only */
# define GLAPI extern
#endif /* _STATIC_MESA support */

#ifdef __cplusplus
extern "C" {
#endif

/*************************************************************/

/* Extensions */
#define GLU_EXT_object_space_tess        1
#define GLU_EXT_nurbs_tessellator        1

/* Boolean */
#define GLU_FALSE                        0
#define GLU_TRUE                         1

/* Version */
#define GLU_VERSION_1_1                  1
#define GLU_VERSION_1_2                  1
#define GLU_VERSION_1_3                  1

/* StringName */
#define GLU_VERSION                      100800
#define GLU_EXTENSIONS                   100801

/* ErrorCode */
#define GLU_INVALID_ENUM                 100900
#define GLU_INVALID_VALUE                100901
#define GLU_OUT_OF_MEMORY                100902
#define GLU_INCOMPATIBLE_GL_VERSION      100903
#define GLU_INVALID_OPERATION            100904

/* NurbsDisplay */
/*      GLU_FILL */
#define GLU_OUTLINE_POLYGON              100240
#define GLU_OUTLINE_PATCH                100241

/* NurbsCallback */
#define GLU_NURBS_ERROR                  100103
#define GLU_ERROR                        100103
#define GLU_NURBS_BEGIN                  100164
```

```
#define GLU_NURBS_BEGIN_EXT             100164
#define GLU_NURBS_VERTEX                100165
#define GLU_NURBS_VERTEX_EXT            100165
#define GLU_NURBS_NORMAL                100166
#define GLU_NURBS_NORMAL_EXT            100166
#define GLU_NURBS_COLOR                 100167
#define GLU_NURBS_COLOR_EXT             100167
#define GLU_NURBS_TEXTURE_COORD         100168
#define GLU_NURBS_TEX_COORD_EXT         100168
#define GLU_NURBS_END                   100169
#define GLU_NURBS_END_EXT               100169
#define GLU_NURBS_BEGIN_DATA            100170
#define GLU_NURBS_BEGIN_DATA_EXT        100170
#define GLU_NURBS_VERTEX_DATA           100171
#define GLU_NURBS_VERTEX_DATA_EXT       100171
#define GLU_NURBS_NORMAL_DATA           100172
#define GLU_NURBS_NORMAL_DATA_EXT       100172
#define GLU_NURBS_COLOR_DATA            100173
#define GLU_NURBS_COLOR_DATA_EXT        100173
#define GLU_NURBS_TEXTURE_COORD_DATA    100174
#define GLU_NURBS_TEX_COORD_DATA_EXT    100174
#define GLU_NURBS_END_DATA              100175
#define GLU_NURBS_END_DATA_EXT          100175

/* NurbsError */
#define GLU_NURBS_ERROR1                100251
#define GLU_NURBS_ERROR2                100252
#define GLU_NURBS_ERROR3                100253
#define GLU_NURBS_ERROR4                100254
#define GLU_NURBS_ERROR5                100255
#define GLU_NURBS_ERROR6                100256
#define GLU_NURBS_ERROR7                100257
#define GLU_NURBS_ERROR8                100258
#define GLU_NURBS_ERROR9                100259
#define GLU_NURBS_ERROR10               100260
#define GLU_NURBS_ERROR11               100261
#define GLU_NURBS_ERROR12               100262
#define GLU_NURBS_ERROR13               100263
#define GLU_NURBS_ERROR14               100264
#define GLU_NURBS_ERROR15               100265
#define GLU_NURBS_ERROR16               100266
#define GLU_NURBS_ERROR17               100267
#define GLU_NURBS_ERROR18               100268
#define GLU_NURBS_ERROR19               100269
#define GLU_NURBS_ERROR20               100270
#define GLU_NURBS_ERROR21               100271
#define GLU_NURBS_ERROR22               100272
#define GLU_NURBS_ERROR23               100273
#define GLU_NURBS_ERROR24               100274
#define GLU_NURBS_ERROR25               100275
#define GLU_NURBS_ERROR26               100276
#define GLU_NURBS_ERROR27               100277
#define GLU_NURBS_ERROR28               100278
#define GLU_NURBS_ERROR29               100279
```

```
#define GLU_NURBS_ERROR30               100280
#define GLU_NURBS_ERROR31               100281
#define GLU_NURBS_ERROR32               100282
#define GLU_NURBS_ERROR33               100283
#define GLU_NURBS_ERROR34               100284
#define GLU_NURBS_ERROR35               100285
#define GLU_NURBS_ERROR36               100286
#define GLU_NURBS_ERROR37               100287

/* NurbsProperty */
#define GLU_AUTO_LOAD_MATRIX            100200
#define GLU_CULLING                     100201
#define GLU_SAMPLING_TOLERANCE          100203
#define GLU_DISPLAY_MODE                100204
#define GLU_PARAMETRIC_TOLERANCE        100202
#define GLU_SAMPLING_METHOD             100205
#define GLU_U_STEP                      100206
#define GLU_V_STEP                      100207
#define GLU_NURBS_MODE                  100160
#define GLU_NURBS_MODE_EXT              100160
#define GLU_NURBS_TESSELLATOR           100161
#define GLU_NURBS_TESSELLATOR_EXT       100161
#define GLU_NURBS_RENDERER              100162
#define GLU_NURBS_RENDERER_EXT          100162

/* NurbsSampling */
#define GLU_OBJECT_PARAMETRIC_ERROR     100208
#define GLU_OBJECT_PARAMETRIC_ERROR_EXT 100208
#define GLU_OBJECT_PATH_LENGTH          100209
#define GLU_OBJECT_PATH_LENGTH_EXT      100209
#define GLU_PATH_LENGTH                 100215
#define GLU_PARAMETRIC_ERROR            100216
#define GLU_DOMAIN_DISTANCE             100217

/* NurbsTrim */
#define GLU_MAP1_TRIM_2                 100210
#define GLU_MAP1_TRIM_3                 100211

/* QuadricDrawStyle */
#define GLU_POINT                       100010
#define GLU_LINE                        100011
#define GLU_FILL                        100012
#define GLU_SILHOUETTE                  100013

/* QuadricCallback */
/*      GLU_ERROR */

/* QuadricNormal */
#define GLU_SMOOTH                      100000
#define GLU_FLAT                        100001
#define GLU_NONE                        100002

/* QuadricOrientation */
#define GLU_OUTSIDE                     100020
```

```
#define GLU_INSIDE                  100021

/* TessCallback */
#define GLU_TESS_BEGIN              100100
#define GLU_BEGIN                   100100
#define GLU_TESS_VERTEX             100101
#define GLU_VERTEX                  100101
#define GLU_TESS_END                100102
#define GLU_END                     100102
#define GLU_TESS_ERROR              100103
#define GLU_TESS_EDGE_FLAG          100104
#define GLU_EDGE_FLAG               100104
#define GLU_TESS_COMBINE            100105
#define GLU_TESS_BEGIN_DATA         100106
#define GLU_TESS_VERTEX_DATA        100107
#define GLU_TESS_END_DATA           100108
#define GLU_TESS_ERROR_DATA         100109
#define GLU_TESS_EDGE_FLAG_DATA     100110
#define GLU_TESS_COMBINE_DATA       100111

/* TessContour */
#define GLU_CW                      100120
#define GLU_CCW                     100121
#define GLU_INTERIOR                100122
#define GLU_EXTERIOR                100123
#define GLU_UNKNOWN                 100124

/* TessProperty */
#define GLU_TESS_WINDING_RULE       100140
#define GLU_TESS_BOUNDARY_ONLY      100141
#define GLU_TESS_TOLERANCE          100142

/* TessError */
#define GLU_TESS_ERROR1             100151
#define GLU_TESS_ERROR2             100152
#define GLU_TESS_ERROR3             100153
#define GLU_TESS_ERROR4             100154
#define GLU_TESS_ERROR5             100155
#define GLU_TESS_ERROR6             100156
#define GLU_TESS_ERROR7             100157
#define GLU_TESS_ERROR8             100158
#define GLU_TESS_MISSING_BEGIN_POLYGON  100151
#define GLU_TESS_MISSING_BEGIN_CONTOUR  100152
#define GLU_TESS_MISSING_END_POLYGON    100153
#define GLU_TESS_MISSING_END_CONTOUR    100154
#define GLU_TESS_COORD_TOO_LARGE        100155
#define GLU_TESS_NEED_COMBINE_CALLBACK  100156

/* TessWinding */
#define GLU_TESS_WINDING_ODD        100130
#define GLU_TESS_WINDING_NONZERO    100131
#define GLU_TESS_WINDING_POSITIVE   100132
#define GLU_TESS_WINDING_NEGATIVE   100133
#define GLU_TESS_WINDING_ABS_GEQ_TWO    100134
```

```
/*************************************************************/


#ifdef __cplusplus
class GLUnurbs;
class GLUquadric;
class GLUtesselator;
#else
typedef struct GLUnurbs GLUnurbs;
typedef struct GLUquadric GLUquadric;
typedef struct GLUtesselator GLUtesselator;
#endif

typedef GLUnurbs GLUnurbsObj;
typedef GLUquadric GLUquadricObj;
typedef GLUtesselator GLUtesselatorObj;
typedef GLUtesselator GLUtriangulatorObj;

#define GLU_TESS_MAX_COORD 1.0e150

/* Internal convenience typedefs */
typedef void (GLAPIENTRYP _GLUfuncptr)(void);

GLAPI void GLAPIENTRY gluBeginCurve (GLUnurbs* nurb);
GLAPI void GLAPIENTRY gluBeginPolygon (GLUtesselator* tess);
GLAPI void GLAPIENTRY gluBeginSurface (GLUnurbs* nurb);
GLAPI void GLAPIENTRY gluBeginTrim (GLUnurbs* nurb);
GLAPI GLint GLAPIENTRY gluBuild1DMipmapLevels (GLenum target, GLint internalFor
mat, GLsizei width, GLenum format, GLenum type, GLint level, GLint base, GLint max, con
st void *data);
GLAPI GLint GLAPIENTRY gluBuild1DMipmaps (GLenum target, GLint internalFormat,
GLsizei width, GLenum format, GLenum type, const void *data);
GLAPI GLint GLAPIENTRY gluBuild2DMipmapLevels (GLenum target, GLint internalFor
mat, GLsizei width, GLsizei height, GLenum format, GLenum type, GLint level, GLint base,
GLint max, const void *data);
GLAPI GLint GLAPIENTRY gluBuild2DMipmaps (GLenum target, GLint internalFormat,
GLsizei width, GLsizei height, GLenum format, GLenum type, const void *data);
GLAPI GLint GLAPIENTRY gluBuild3DMipmapLevels (GLenum target, GLint internalFor
mat, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLenum type, GLint lev
el, GLint base, GLint max, const void *data);
GLAPI GLint GLAPIENTRY gluBuild3DMipmaps (GLenum target, GLint internalFormat,
GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLenum type, const void *da
ta);
GLAPI GLboolean GLAPIENTRY gluCheckExtension (const GLubyte *extName, const GL
ubyte *extString);
GLAPI void GLAPIENTRY gluCylinder (GLUquadric* quad, GLdouble base, GLdouble to
p, GLdouble height, GLint slices, GLint stacks);
GLAPI void GLAPIENTRY gluDeleteNurbsRenderer (GLUnurbs* nurb);
GLAPI void GLAPIENTRY gluDeleteQuadric (GLUquadric* quad);
GLAPI void GLAPIENTRY gluDeleteTess (GLUtesselator* tess);
GLAPI void GLAPIENTRY gluDisk (GLUquadric* quad, GLdouble inner, GLdouble outer,
GLint slices, GLint loops);
GLAPI void GLAPIENTRY gluEndCurve (GLUnurbs* nurb);
```

```
GLAPI void GLAPIENTRY gluEndPolygon (GLUtesselator* tess);
GLAPI void GLAPIENTRY gluEndSurface (GLUnurbs* nurb);
GLAPI void GLAPIENTRY gluEndTrim (GLUnurbs* nurb);
GLAPI const GLubyte * GLAPIENTRY gluErrorString (GLenum error);
GLAPI void GLAPIENTRY gluGetNurbsProperty (GLUnurbs* nurb, GLenum property, GLf
loat* data);
GLAPI const GLubyte * GLAPIENTRY gluGetString (GLenum name);
GLAPI void GLAPIENTRY gluGetTessProperty (GLUtesselator* tess, GLenum which, GLd
ouble* data);
GLAPI void GLAPIENTRY gluLoadSamplingMatrices (GLUnurbs* nurb, const GLfloat *m
odel, const GLfloat *perspective, const GLint *view);
GLAPI void GLAPIENTRY gluLookAt (GLdouble eyeX, GLdouble eyeY, GLdouble eyeZ,
GLdouble centerX, GLdouble centerY, GLdouble centerZ, GLdouble upX, GLdouble upY, G
Ldouble upZ);
GLAPI GLUnurbs* GLAPIENTRY gluNewNurbsRenderer (void);
GLAPI GLUquadric* GLAPIENTRY gluNewQuadric (void);
GLAPI GLUtesselator* GLAPIENTRY gluNewTess (void);
GLAPI void GLAPIENTRY gluNextContour (GLUtesselator* tess, GLenum type);
GLAPI void GLAPIENTRY gluNurbsCallback (GLUnurbs* nurb, GLenum which, _GLUfun
cptr CallBackFunc);
GLAPI void GLAPIENTRY gluNurbsCallbackData (GLUnurbs* nurb, GLvoid* userData);
GLAPI void GLAPIENTRY gluNurbsCallbackDataEXT (GLUnurbs* nurb, GLvoid* userDa
ta);
GLAPI void GLAPIENTRY gluNurbsCurve (GLUnurbs* nurb, GLint knotCount, GLfloat *k
nots, GLint stride, GLfloat *control, GLint order, GLenum type);
GLAPI void GLAPIENTRY gluNurbsProperty (GLUnurbs* nurb, GLenum property, GLfloa
t value);
GLAPI void GLAPIENTRY gluNurbsSurface (GLUnurbs* nurb, GLint sKnotCount, GLfloat
* sKnots, GLint tKnotCount, GLfloat* tKnots, GLint sStride, GLint tStride, GLfloat* contro
l, GLint sOrder, GLint tOrder, GLenum type);
GLAPI void GLAPIENTRY gluOrtho2D (GLdouble left, GLdouble right, GLdouble bottom,
GLdouble top);
GLAPI void GLAPIENTRY gluPartialDisk (GLUquadric* quad, GLdouble inner, GLdouble
outer, GLint slices, GLint loops, GLdouble start, GLdouble sweep);
GLAPI void GLAPIENTRY gluPerspective (GLdouble fovy, GLdouble aspect, GLdouble zN
ear, GLdouble zFar);
GLAPI void GLAPIENTRY gluPickMatrix (GLdouble x, GLdouble y, GLdouble delX, GLd
ouble delY, GLint *viewport);
GLAPI GLint GLAPIENTRY gluProject (GLdouble objX, GLdouble objY, GLdouble objZ, c
onst GLdouble *model, const GLdouble *proj, const GLint *view, GLdouble* winX, GLdou
ble* winY, GLdouble* winZ);
GLAPI void GLAPIENTRY gluPwlCurve (GLUnurbs* nurb, GLint count, GLfloat* data, G
Lint stride, GLenum type);
GLAPI void GLAPIENTRY gluQuadricCallback (GLUquadric* quad, GLenum which, _GL
Ufuncptr CallBackFunc);
GLAPI void GLAPIENTRY gluQuadricDrawStyle (GLUquadric* quad, GLenum draw);
GLAPI void GLAPIENTRY gluQuadricNormals (GLUquadric* quad, GLenum normal);
GLAPI void GLAPIENTRY gluQuadricOrientation (GLUquadric* quad, GLenum orientatio
n);
GLAPI void GLAPIENTRY gluQuadricTexture (GLUquadric* quad, GLboolean texture);
GLAPI GLint GLAPIENTRY gluScaleImage (GLenum format, GLsizei wIn, GLsizei hIn, G
Lenum typeIn, const void *dataIn, GLsizei wOut, GLsizei hOut, GLenum typeOut, GLvoid*
dataOut);
GLAPI void GLAPIENTRY gluSphere (GLUquadric* quad, GLdouble radius, GLint slices,
```

```
GLint stacks);
GLAPI void GLAPIENTRY gluTessBeginContour (GLUtesselator* tess);
GLAPI void GLAPIENTRY gluTessBeginPolygon (GLUtesselator* tess, GLvoid* data);
GLAPI void GLAPIENTRY gluTessCallback (GLUtesselator* tess, GLenum which, _GLUfu
ncptr CallBackFunc);
GLAPI void GLAPIENTRY gluTessEndContour (GLUtesselator* tess);
GLAPI void GLAPIENTRY gluTessEndPolygon (GLUtesselator* tess);
GLAPI void GLAPIENTRY gluTessNormal (GLUtesselator* tess, GLdouble valueX, GLdou
ble valueY, GLdouble valueZ);
GLAPI void GLAPIENTRY gluTessProperty (GLUtesselator* tess, GLenum which, GLdoub
le data);
GLAPI void GLAPIENTRY gluTessVertex (GLUtesselator* tess, GLdouble *location, GLvo
id* data);
GLAPI GLint GLAPIENTRY gluUnProject (GLdouble winX, GLdouble winY, GLdouble wi
nZ, const GLdouble *model, const GLdouble *proj, const GLint *view, GLdouble* objX, G
Ldouble* objY, GLdouble* objZ);
GLAPI GLint GLAPIENTRY gluUnProject4 (GLdouble winX, GLdouble winY, GLdouble w
inZ, GLdouble clipW, const GLdouble *model, const GLdouble *proj, const GLint *view, G
Ldouble nearVal, GLdouble farVal, GLdouble* objX, GLdouble* objY, GLdouble* objZ, GL
double* objW);

#ifdef __cplusplus
}
#endif

#endif /* __glu_h__ */
```