```
#ifndef  __FREEGLUT_STD_H__
#define  __FREEGLUT_STD_H__

/*
 * freeglut_std.h
 *
 * The GLUT-compatible part of the freeglut library include file
 *
 * Copyright (c) 1999-2000 Pawel W. Olszta. All Rights Reserved.
 * Written by Pawel W. Olszta, <olszta@sourceforge.net>
 * Creation date: Thu Dec 2 1999
 *
 * Permission is hereby granted, free of charge, to any person obtaining a
 * copy of this software and associated documentation files (the "Software"),
 * to deal in the Software without restriction, including without limitation
 * the rights to use, copy, modify, merge, publish, distribute, sublicense,
 * and/or sell copies of the Software, and to permit persons to whom the
 * Software is furnished to do so, subject to the following conditions:
 *
 * The above copyright notice and this permission notice shall be included
 * in all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, E
XPRESS
 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCH
ANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  IN NO EVE
NT SHALL
 * PAWEL W. OLSZTA BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILI
TY, WHETHER
 * IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
OR IN
 * CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN TH
E SOFTWARE.
 */

#ifdef __cplusplus
   extern "C" {
#endif

/*
 * Under windows, we have to differentiate between static and dynamic libraries
 */
#ifdef _WIN32
/* #pragma may not be supported by some compilers.
 * Discussion by FreeGLUT developers suggests that
 * Visual C++ specific code involving pragmas may
 * need to move to a separate header.  24th Dec 2003
 */

/* Define FREEGLUT_LIB_PRAGMAS to 1 to include library
 * pragmas or to 0 to exclude library pragmas.
 * The default behavior depends on the compiler/platform.
 */
```

```
#  ifndef FREEGLUT_LIB_PRAGMAS
#     if ( defined(_MSC_VER) || defined(__WATCOMC__) ) && !defined(_WIN32_WCE)
#        define FREEGLUT_LIB_PRAGMAS 1
#     else
#        define FREEGLUT_LIB_PRAGMAS 0
#     endif
#  endif

# ifndef WIN32_LEAN_AND_MEAN
#   define WIN32_LEAN_AND_MEAN 1
# endif
# ifndef NOMINMAX
#   define NOMINMAX
# endif
#  include <windows.h>

/* Windows static library */
#  ifdef FREEGLUT_STATIC

#     define FGAPI
#     define FGAPIENTRY

      /* Link with Win32 static freeglut lib */
#     if FREEGLUT_LIB_PRAGMAS
#        pragma comment (lib, "freeglut_static.lib")
#     endif

/* Windows shared library (DLL) */
#  else

#     define FGAPIENTRY __stdcall
#     if defined(FREEGLUT_EXPORTS)
#        define FGAPI __declspec(dllexport)
#     else
#        define FGAPI __declspec(dllimport)

         /* Link with Win32 shared freeglut lib */
#        if FREEGLUT_LIB_PRAGMAS
#           pragma comment (lib, "freeglut.lib")
#        endif

#     endif

#  endif

/* Drag in other Windows libraries as required by FreeGLUT */
#  if FREEGLUT_LIB_PRAGMAS
#     pragma comment (lib, "glu32.lib")    /* link OpenGL Utility lib     */
#     pragma comment (lib, "opengl32.lib") /* link Microsoft OpenGL lib   */
#     pragma comment (lib, "gdi32.lib")    /* link Windows GDI lib        */
#     pragma comment (lib, "winmm.lib")    /* link Windows MultiMedia lib */
#     pragma comment (lib, "user32.lib")   /* link Windows user lib       */
#  endif
```

```
#else

/* Non-Windows definition of FGAPI and FGAPIENTRY  */
#       define FGAPI
#       define FGAPIENTRY

#endif

/*
 * The freeglut and GLUT API versions
 */
#define  FREEGLUT           1
#define  GLUT_API_VERSION    4
#define  GLUT_XLIB_IMPLEMENTATION 13
/* Deprecated:
   cf. http://sourceforge.net/mailarchive/forum.php?thread_name=CABcAi1hw7cr4xtigckaGXB5X8wddLfMcbA_rZ3NAuwMrX_zmsw%40mail.gmail.com&forum_name=freeglut-developer */
#define  FREEGLUT_VERSION_2_0 1

/*
 * Always include OpenGL and GLU headers
 */
#if __APPLE__
#   include <OpenGL/gl.h>
#   include <OpenGL/glu.h>
#else
#   include <GL/gl.h>
#   include <GL/glu.h>
#endif

/*
 * GLUT API macro definitions -- the special key codes:
 */
#define  GLUT_KEY_F1                        0x0001
#define  GLUT_KEY_F2                        0x0002
#define  GLUT_KEY_F3                        0x0003
#define  GLUT_KEY_F4                        0x0004
#define  GLUT_KEY_F5                        0x0005
#define  GLUT_KEY_F6                        0x0006
#define  GLUT_KEY_F7                        0x0007
#define  GLUT_KEY_F8                        0x0008
#define  GLUT_KEY_F9                        0x0009
#define  GLUT_KEY_F10                       0x000A
#define  GLUT_KEY_F11                       0x000B
#define  GLUT_KEY_F12                       0x000C
#define  GLUT_KEY_LEFT                      0x0064
#define  GLUT_KEY_UP                        0x0065
#define  GLUT_KEY_RIGHT                     0x0066
#define  GLUT_KEY_DOWN                      0x0067
#define  GLUT_KEY_PAGE_UP                   0x0068
#define  GLUT_KEY_PAGE_DOWN                 0x0069
#define  GLUT_KEY_HOME                      0x006A
#define  GLUT_KEY_END                       0x006B
```

```
#define  GLUT_KEY_INSERT                0x006C

/*
 * GLUT API macro definitions -- mouse state definitions
 */
#define  GLUT_LEFT_BUTTON               0x0000
#define  GLUT_MIDDLE_BUTTON             0x0001
#define  GLUT_RIGHT_BUTTON              0x0002
#define  GLUT_DOWN                      0x0000
#define  GLUT_UP                        0x0001
#define  GLUT_LEFT                      0x0000
#define  GLUT_ENTERED                   0x0001

/*
 * GLUT API macro definitions -- the display mode definitions
 */
#define  GLUT_RGB                       0x0000
#define  GLUT_RGBA                      0x0000
#define  GLUT_INDEX                     0x0001
#define  GLUT_SINGLE                    0x0000
#define  GLUT_DOUBLE                    0x0002
#define  GLUT_ACCUM                     0x0004
#define  GLUT_ALPHA                     0x0008
#define  GLUT_DEPTH                     0x0010
#define  GLUT_STENCIL                   0x0020
#define  GLUT_MULTISAMPLE               0x0080
#define  GLUT_STEREO                    0x0100
#define  GLUT_LUMINANCE                 0x0200

/*
 * GLUT API macro definitions -- windows and menu related definitions
 */
#define  GLUT_MENU_NOT_IN_USE           0x0000
#define  GLUT_MENU_IN_USE               0x0001
#define  GLUT_NOT_VISIBLE               0x0000
#define  GLUT_VISIBLE                   0x0001
#define  GLUT_HIDDEN                    0x0000
#define  GLUT_FULLY_RETAINED            0x0001
#define  GLUT_PARTIALLY_RETAINED        0x0002
#define  GLUT_FULLY_COVERED             0x0003

/*
 * GLUT API macro definitions -- fonts definitions
 *
 * Steve Baker suggested to make it binary compatible with GLUT:
 */
#if defined(_MSC_VER) || defined(__CYGWIN__) || defined(__MINGW32__) || defined(__
WATCOMC__)
#   define  GLUT_STROKE_ROMAN           ((void *)0x0000)
#   define  GLUT_STROKE_MONO_ROMAN      ((void *)0x0001)
#   define  GLUT_BITMAP_9_BY_15         ((void *)0x0002)
#   define  GLUT_BITMAP_8_BY_13         ((void *)0x0003)
#   define  GLUT_BITMAP_TIMES_ROMAN_10  ((void *)0x0004)
#   define  GLUT_BITMAP_TIMES_ROMAN_24  ((void *)0x0005)
```

```
#  define  GLUT_BITMAP_HELVETICA_10       ((void *)0x0006)
#  define  GLUT_BITMAP_HELVETICA_12       ((void *)0x0007)
#  define  GLUT_BITMAP_HELVETICA_18       ((void *)0x0008)
#else
   /*
    * I don't really know if it's a good idea... But here it goes:
    */
   extern void* glutStrokeRoman;
   extern void* glutStrokeMonoRoman;
   extern void* glutBitmap9By15;
   extern void* glutBitmap8By13;
   extern void* glutBitmapTimesRoman10;
   extern void* glutBitmapTimesRoman24;
   extern void* glutBitmapHelvetica10;
   extern void* glutBitmapHelvetica12;
   extern void* glutBitmapHelvetica18;

   /*
    * Those pointers will be used by following definitions:
    */
#  define  GLUT_STROKE_ROMAN              ((void *) &glutStrokeRoman)
#  define  GLUT_STROKE_MONO_ROMAN         ((void *) &glutStrokeMonoRoman)
#  define  GLUT_BITMAP_9_BY_15            ((void *) &glutBitmap9By15)
#  define  GLUT_BITMAP_8_BY_13            ((void *) &glutBitmap8By13)
#  define  GLUT_BITMAP_TIMES_ROMAN_10     ((void *) &glutBitmapTimesRoman10)
#  define  GLUT_BITMAP_TIMES_ROMAN_24     ((void *) &glutBitmapTimesRoman24)
#  define  GLUT_BITMAP_HELVETICA_10       ((void *) &glutBitmapHelvetica10)
#  define  GLUT_BITMAP_HELVETICA_12       ((void *) &glutBitmapHelvetica12)
#  define  GLUT_BITMAP_HELVETICA_18       ((void *) &glutBitmapHelvetica18)
#endif

/*
 * GLUT API macro definitions -- the glutGet parameters
 */
#define  GLUT_WINDOW_X                  0x0064
#define  GLUT_WINDOW_Y                  0x0065
#define  GLUT_WINDOW_WIDTH              0x0066
#define  GLUT_WINDOW_HEIGHT             0x0067
#define  GLUT_WINDOW_BUFFER_SIZE        0x0068
#define  GLUT_WINDOW_STENCIL_SIZE       0x0069
#define  GLUT_WINDOW_DEPTH_SIZE         0x006A
#define  GLUT_WINDOW_RED_SIZE           0x006B
#define  GLUT_WINDOW_GREEN_SIZE         0x006C
#define  GLUT_WINDOW_BLUE_SIZE          0x006D
#define  GLUT_WINDOW_ALPHA_SIZE         0x006E
#define  GLUT_WINDOW_ACCUM_RED_SIZE     0x006F
#define  GLUT_WINDOW_ACCUM_GREEN_SIZE   0x0070
#define  GLUT_WINDOW_ACCUM_BLUE_SIZE    0x0071
#define  GLUT_WINDOW_ACCUM_ALPHA_SIZE   0x0072
#define  GLUT_WINDOW_DOUBLEBUFFER       0x0073
#define  GLUT_WINDOW_RGBA               0x0074
#define  GLUT_WINDOW_PARENT             0x0075
#define  GLUT_WINDOW_NUM_CHILDREN       0x0076
#define  GLUT_WINDOW_COLORMAP_SIZE      0x0077
```

```
#define  GLUT_WINDOW_NUM_SAMPLES         0x0078
#define  GLUT_WINDOW_STEREO              0x0079
#define  GLUT_WINDOW_CURSOR              0x007A

#define  GLUT_SCREEN_WIDTH               0x00C8
#define  GLUT_SCREEN_HEIGHT              0x00C9
#define  GLUT_SCREEN_WIDTH_MM            0x00CA
#define  GLUT_SCREEN_HEIGHT_MM           0x00CB
#define  GLUT_MENU_NUM_ITEMS             0x012C
#define  GLUT_DISPLAY_MODE_POSSIBLE      0x0190
#define  GLUT_INIT_WINDOW_X              0x01F4
#define  GLUT_INIT_WINDOW_Y              0x01F5
#define  GLUT_INIT_WINDOW_WIDTH          0x01F6
#define  GLUT_INIT_WINDOW_HEIGHT         0x01F7
#define  GLUT_INIT_DISPLAY_MODE          0x01F8
#define  GLUT_ELAPSED_TIME               0x02BC
#define  GLUT_WINDOW_FORMAT_ID           0x007B

/*
 * GLUT API macro definitions -- the glutDeviceGet parameters
 */
#define  GLUT_HAS_KEYBOARD               0x0258
#define  GLUT_HAS_MOUSE                  0x0259
#define  GLUT_HAS_SPACEBALL              0x025A
#define  GLUT_HAS_DIAL_AND_BUTTON_BOX    0x025B
#define  GLUT_HAS_TABLET                 0x025C
#define  GLUT_NUM_MOUSE_BUTTONS          0x025D
#define  GLUT_NUM_SPACEBALL_BUTTONS      0x025E
#define  GLUT_NUM_BUTTON_BOX_BUTTONS     0x025F
#define  GLUT_NUM_DIALS                  0x0260
#define  GLUT_NUM_TABLET_BUTTONS         0x0261
#define  GLUT_DEVICE_IGNORE_KEY_REPEAT   0x0262
#define  GLUT_DEVICE_KEY_REPEAT          0x0263
#define  GLUT_HAS_JOYSTICK               0x0264
#define  GLUT_OWNS_JOYSTICK              0x0265
#define  GLUT_JOYSTICK_BUTTONS           0x0266
#define  GLUT_JOYSTICK_AXES              0x0267
#define  GLUT_JOYSTICK_POLL_RATE         0x0268

/*
 * GLUT API macro definitions -- the glutLayerGet parameters
 */
#define  GLUT_OVERLAY_POSSIBLE           0x0320
#define  GLUT_LAYER_IN_USE               0x0321
#define  GLUT_HAS_OVERLAY                0x0322
#define  GLUT_TRANSPARENT_INDEX          0x0323
#define  GLUT_NORMAL_DAMAGED             0x0324
#define  GLUT_OVERLAY_DAMAGED            0x0325

/*
 * GLUT API macro definitions -- the glutVideoResizeGet parameters
 */
#define  GLUT_VIDEO_RESIZE_POSSIBLE      0x0384
#define  GLUT_VIDEO_RESIZE_IN_USE        0x0385
```

```
#define  GLUT_VIDEO_RESIZE_X_DELTA        0x0386
#define  GLUT_VIDEO_RESIZE_Y_DELTA        0x0387
#define  GLUT_VIDEO_RESIZE_WIDTH_DELTA    0x0388
#define  GLUT_VIDEO_RESIZE_HEIGHT_DELTA   0x0389
#define  GLUT_VIDEO_RESIZE_X              0x038A
#define  GLUT_VIDEO_RESIZE_Y              0x038B
#define  GLUT_VIDEO_RESIZE_WIDTH          0x038C
#define  GLUT_VIDEO_RESIZE_HEIGHT         0x038D

/*
 * GLUT API macro definitions -- the glutUseLayer parameters
 */
#define  GLUT_NORMAL                      0x0000
#define  GLUT_OVERLAY                     0x0001

/*
 * GLUT API macro definitions -- the glutGetModifiers parameters
 */
#define  GLUT_ACTIVE_SHIFT                0x0001
#define  GLUT_ACTIVE_CTRL                 0x0002
#define  GLUT_ACTIVE_ALT                  0x0004

/*
 * GLUT API macro definitions -- the glutSetCursor parameters
 */
#define  GLUT_CURSOR_RIGHT_ARROW          0x0000
#define  GLUT_CURSOR_LEFT_ARROW           0x0001
#define  GLUT_CURSOR_INFO                 0x0002
#define  GLUT_CURSOR_DESTROY              0x0003
#define  GLUT_CURSOR_HELP                 0x0004
#define  GLUT_CURSOR_CYCLE                0x0005
#define  GLUT_CURSOR_SPRAY                0x0006
#define  GLUT_CURSOR_WAIT                 0x0007
#define  GLUT_CURSOR_TEXT                 0x0008
#define  GLUT_CURSOR_CROSSHAIR            0x0009
#define  GLUT_CURSOR_UP_DOWN              0x000A
#define  GLUT_CURSOR_LEFT_RIGHT           0x000B
#define  GLUT_CURSOR_TOP_SIDE             0x000C
#define  GLUT_CURSOR_BOTTOM_SIDE          0x000D
#define  GLUT_CURSOR_LEFT_SIDE            0x000E
#define  GLUT_CURSOR_RIGHT_SIDE           0x000F
#define  GLUT_CURSOR_TOP_LEFT_CORNER      0x0010
#define  GLUT_CURSOR_TOP_RIGHT_CORNER     0x0011
#define  GLUT_CURSOR_BOTTOM_RIGHT_CORNER  0x0012
#define  GLUT_CURSOR_BOTTOM_LEFT_CORNER   0x0013
#define  GLUT_CURSOR_INHERIT              0x0064
#define  GLUT_CURSOR_NONE                 0x0065
#define  GLUT_CURSOR_FULL_CROSSHAIR       0x0066

/*
 * GLUT API macro definitions -- RGB color component specification definitions
 */
#define  GLUT_RED                         0x0000
#define  GLUT_GREEN                       0x0001
```

```
#define  GLUT_BLUE                      0x0002

/*
 * GLUT API macro definitions -- additional keyboard and joystick definitions
 */
#define  GLUT_KEY_REPEAT_OFF             0x0000
#define  GLUT_KEY_REPEAT_ON              0x0001
#define  GLUT_KEY_REPEAT_DEFAULT         0x0002

#define  GLUT_JOYSTICK_BUTTON_A          0x0001
#define  GLUT_JOYSTICK_BUTTON_B          0x0002
#define  GLUT_JOYSTICK_BUTTON_C          0x0004
#define  GLUT_JOYSTICK_BUTTON_D          0x0008

/*
 * GLUT API macro definitions -- game mode definitions
 */
#define  GLUT_GAME_MODE_ACTIVE           0x0000
#define  GLUT_GAME_MODE_POSSIBLE         0x0001
#define  GLUT_GAME_MODE_WIDTH            0x0002
#define  GLUT_GAME_MODE_HEIGHT           0x0003
#define  GLUT_GAME_MODE_PIXEL_DEPTH      0x0004
#define  GLUT_GAME_MODE_REFRESH_RATE     0x0005
#define  GLUT_GAME_MODE_DISPLAY_CHANGED  0x0006

/*
 * Initialization functions, see fglut_init.c
 */
FGAPI void   FGAPIENTRY glutInit( int* pargc, char** argv );
FGAPI void   FGAPIENTRY glutInitWindowPosition( int x, int y );
FGAPI void   FGAPIENTRY glutInitWindowSize( int width, int height );
FGAPI void   FGAPIENTRY glutInitDisplayMode( unsigned int displayMode );
FGAPI void   FGAPIENTRY glutInitDisplayString( const char* displayMode );

/*
 * Process loop function, see freeglut_main.c
 */
FGAPI void   FGAPIENTRY glutMainLoop( void );

/*
 * Window management functions, see freeglut_window.c
 */
FGAPI int    FGAPIENTRY glutCreateWindow( const char* title );
FGAPI int    FGAPIENTRY glutCreateSubWindow( int window, int x, int y, int width, int
height );
FGAPI void   FGAPIENTRY glutDestroyWindow( int window );
FGAPI void   FGAPIENTRY glutSetWindow( int window );
FGAPI int    FGAPIENTRY glutGetWindow( void );
FGAPI void   FGAPIENTRY glutSetWindowTitle( const char* title );
FGAPI void   FGAPIENTRY glutSetIconTitle( const char* title );
FGAPI void   FGAPIENTRY glutReshapeWindow( int width, int height );
FGAPI void   FGAPIENTRY glutPositionWindow( int x, int y );
FGAPI void   FGAPIENTRY glutShowWindow( void );
FGAPI void   FGAPIENTRY glutHideWindow( void );
```

```
FGAPI void   FGAPIENTRY glutIconifyWindow( void );
FGAPI void   FGAPIENTRY glutPushWindow( void );
FGAPI void   FGAPIENTRY glutPopWindow( void );
FGAPI void   FGAPIENTRY glutFullScreen( void );

/*
 * Display-connected functions, see freeglut_display.c
 */
FGAPI void   FGAPIENTRY glutPostWindowRedisplay( int window );
FGAPI void   FGAPIENTRY glutPostRedisplay( void );
FGAPI void   FGAPIENTRY glutSwapBuffers( void );

/*
 * Mouse cursor functions, see freeglut_cursor.c
 */
FGAPI void   FGAPIENTRY glutWarpPointer( int x, int y );
FGAPI void   FGAPIENTRY glutSetCursor( int cursor );

/*
 * Overlay stuff, see freeglut_overlay.c
 */
FGAPI void   FGAPIENTRY glutEstablishOverlay( void );
FGAPI void   FGAPIENTRY glutRemoveOverlay( void );
FGAPI void   FGAPIENTRY glutUseLayer( GLenum layer );
FGAPI void   FGAPIENTRY glutPostOverlayRedisplay( void );
FGAPI void   FGAPIENTRY glutPostWindowOverlayRedisplay( int window );
FGAPI void   FGAPIENTRY glutShowOverlay( void );
FGAPI void   FGAPIENTRY glutHideOverlay( void );

/*
 * Menu stuff, see freeglut_menu.c
 */
FGAPI int    FGAPIENTRY glutCreateMenu( void (* callback)( int menu ) );
FGAPI void   FGAPIENTRY glutDestroyMenu( int menu );
FGAPI int    FGAPIENTRY glutGetMenu( void );
FGAPI void   FGAPIENTRY glutSetMenu( int menu );
FGAPI void   FGAPIENTRY glutAddMenuEntry( const char* label, int value );
FGAPI void   FGAPIENTRY glutAddSubMenu( const char* label, int subMenu );
FGAPI void   FGAPIENTRY glutChangeToMenuEntry( int item, const char* label, int valu
e );
FGAPI void   FGAPIENTRY glutChangeToSubMenu( int item, const char* label, int value
);
FGAPI void   FGAPIENTRY glutRemoveMenuItem( int item );
FGAPI void   FGAPIENTRY glutAttachMenu( int button );
FGAPI void   FGAPIENTRY glutDetachMenu( int button );

/*
 * Global callback functions, see freeglut_callbacks.c
 */
FGAPI void   FGAPIENTRY glutTimerFunc( unsigned int time, void (* callback)( int ), int
value );
FGAPI void   FGAPIENTRY glutIdleFunc( void (* callback)( void ) );

/*
```

```
 * Window-specific callback functions, see freeglut_callbacks.c
 */
FGAPI void    FGAPIENTRY glutKeyboardFunc( void (* callback)( unsigned char, int, int
) );
FGAPI void    FGAPIENTRY glutSpecialFunc( void (* callback)( int, int, int ) );
FGAPI void    FGAPIENTRY glutReshapeFunc( void (* callback)( int, int ) );
FGAPI void    FGAPIENTRY glutVisibilityFunc( void (* callback)( int ) );
FGAPI void    FGAPIENTRY glutDisplayFunc( void (* callback)( void ) );
FGAPI void    FGAPIENTRY glutMouseFunc( void (* callback)( int, int, int, int ) );
FGAPI void    FGAPIENTRY glutMotionFunc( void (* callback)( int, int ) );
FGAPI void    FGAPIENTRY glutPassiveMotionFunc( void (* callback)( int, int ) );
FGAPI void    FGAPIENTRY glutEntryFunc( void (* callback)( int ) );

FGAPI void    FGAPIENTRY glutKeyboardUpFunc( void (* callback)( unsigned char, int, i
nt ) );
FGAPI void    FGAPIENTRY glutSpecialUpFunc( void (* callback)( int, int, int ) );
FGAPI void    FGAPIENTRY glutJoystickFunc( void (* callback)( unsigned int, int, int, int
), int pollInterval );
FGAPI void    FGAPIENTRY glutMenuStateFunc( void (* callback)( int ) );
FGAPI void    FGAPIENTRY glutMenuStatusFunc( void (* callback)( int, int, int ) );
FGAPI void    FGAPIENTRY glutOverlayDisplayFunc( void (* callback)( void ) );
FGAPI void    FGAPIENTRY glutWindowStatusFunc( void (* callback)( int ) );

FGAPI void    FGAPIENTRY glutSpaceballMotionFunc( void (* callback)( int, int, int ) );
FGAPI void    FGAPIENTRY glutSpaceballRotateFunc( void (* callback)( int, int, int ) );
FGAPI void    FGAPIENTRY glutSpaceballButtonFunc( void (* callback)( int, int ) );
FGAPI void    FGAPIENTRY glutButtonBoxFunc( void (* callback)( int, int ) );
FGAPI void    FGAPIENTRY glutDialsFunc( void (* callback)( int, int ) );
FGAPI void    FGAPIENTRY glutTabletMotionFunc( void (* callback)( int, int ) );
FGAPI void    FGAPIENTRY glutTabletButtonFunc( void (* callback)( int, int, int, int ) );

/*
 * State setting and retrieval functions, see freeglut_state.c
 */
FGAPI int     FGAPIENTRY glutGet( GLenum query );
FGAPI int     FGAPIENTRY glutDeviceGet( GLenum query );
FGAPI int     FGAPIENTRY glutGetModifiers( void );
FGAPI int     FGAPIENTRY glutLayerGet( GLenum query );

/*
 * Font stuff, see freeglut_font.c
 */
FGAPI void    FGAPIENTRY glutBitmapCharacter( void* font, int character );
FGAPI int     FGAPIENTRY glutBitmapWidth( void* font, int character );
FGAPI void    FGAPIENTRY glutStrokeCharacter( void* font, int character );
FGAPI int     FGAPIENTRY glutStrokeWidth( void* font, int character );
FGAPI int     FGAPIENTRY glutBitmapLength( void* font, const unsigned char* string );
FGAPI int     FGAPIENTRY glutStrokeLength( void* font, const unsigned char* string );

/*
 * Geometry functions, see freeglut_geometry.c
 */
FGAPI void    FGAPIENTRY glutWireCube( GLdouble size );
FGAPI void    FGAPIENTRY glutSolidCube( GLdouble size );
```

```
FGAPI void   FGAPIENTRY glutWireSphere( GLdouble radius, GLint slices, GLint stacks
);
FGAPI void   FGAPIENTRY glutSolidSphere( GLdouble radius, GLint slices, GLint stacks
);
FGAPI void   FGAPIENTRY glutWireCone( GLdouble base, GLdouble height, GLint slice
s, GLint stacks );
FGAPI void   FGAPIENTRY glutSolidCone( GLdouble base, GLdouble height, GLint slice
s, GLint stacks );

FGAPI void   FGAPIENTRY glutWireTorus( GLdouble innerRadius, GLdouble outerRadiu
s, GLint sides, GLint rings );
FGAPI void   FGAPIENTRY glutSolidTorus( GLdouble innerRadius, GLdouble outerRadiu
s, GLint sides, GLint rings );
FGAPI void   FGAPIENTRY glutWireDodecahedron( void );
FGAPI void   FGAPIENTRY glutSolidDodecahedron( void );
FGAPI void   FGAPIENTRY glutWireOctahedron( void );
FGAPI void   FGAPIENTRY glutSolidOctahedron( void );
FGAPI void   FGAPIENTRY glutWireTetrahedron( void );
FGAPI void   FGAPIENTRY glutSolidTetrahedron( void );
FGAPI void   FGAPIENTRY glutWireIcosahedron( void );
FGAPI void   FGAPIENTRY glutSolidIcosahedron( void );

/*
 * Teapot rendering functions, found in freeglut_teapot.c
 * NB: front facing polygons have clockwise winding, not counter clockwise
 */
FGAPI void   FGAPIENTRY glutWireTeapot( GLdouble size );
FGAPI void   FGAPIENTRY glutSolidTeapot( GLdouble size );

/*
 * Game mode functions, see freeglut_gamemode.c
 */
FGAPI void   FGAPIENTRY glutGameModeString( const char* string );
FGAPI int    FGAPIENTRY glutEnterGameMode( void );
FGAPI void   FGAPIENTRY glutLeaveGameMode( void );
FGAPI int    FGAPIENTRY glutGameModeGet( GLenum query );

/*
 * Video resize functions, see freeglut_videoresize.c
 */
FGAPI int    FGAPIENTRY glutVideoResizeGet( GLenum query );
FGAPI void   FGAPIENTRY glutSetupVideoResizing( void );
FGAPI void   FGAPIENTRY glutStopVideoResizing( void );
FGAPI void   FGAPIENTRY glutVideoResize( int x, int y, int width, int height );
FGAPI void   FGAPIENTRY glutVideoPan( int x, int y, int width, int height );

/*
 * Colormap functions, see freeglut_misc.c
 */
FGAPI void   FGAPIENTRY glutSetColor( int color, GLfloat red, GLfloat green, GLfloat bl
ue );
FGAPI GLfloat FGAPIENTRY glutGetColor( int color, int component );
FGAPI void   FGAPIENTRY glutCopyColormap( int window );
```

```
/*
 * Misc keyboard and joystick functions, see freeglut_misc.c
 */
FGAPI void    FGAPIENTRY glutIgnoreKeyRepeat( int ignore );
FGAPI void    FGAPIENTRY glutSetKeyRepeat( int repeatMode );
FGAPI void    FGAPIENTRY glutForceJoystickFunc( void );

/*
 * Misc functions, see freeglut_misc.c
 */
FGAPI int     FGAPIENTRY glutExtensionSupported( const char* extension );
FGAPI void    FGAPIENTRY glutReportErrors( void );

/* Comment from glut.h of classic GLUT:

   Win32 has an annoying issue where there are multiple C run-time
   libraries (CRTs).  If the executable is linked with a different CRT
   from the GLUT DLL, the GLUT DLL will not share the same CRT static
   data seen by the executable.  In particular, atexit callbacks registered
   in the executable will not be called if GLUT calls its (different)
   exit routine).  GLUT is typically built with the
   "/MD" option (the CRT with multithreading DLL support), but the Visual
   C++ linker default is "/ML" (the single threaded CRT).

   One workaround to this issue is requiring users to always link with
   the same CRT as GLUT is compiled with.  That requires users supply a
   non-standard option.  GLUT 3.7 has its own built-in workaround where
   the executable's "exit" function pointer is covertly passed to GLUT.
   GLUT then calls the executable's exit function pointer to ensure that
   any "atexit" calls registered by the application are called if GLUT
   needs to exit.

   Note that the __glut*WithExit routines should NEVER be called directly.
   To avoid the atexit workaround, #define GLUT_DISABLE_ATEXIT_HACK. */

/* to get the prototype for exit() */
#include <stdlib.h>

#if defined(_WIN32) && !defined(GLUT_DISABLE_ATEXIT_HACK) && !defined(__WA
TCOMC__)
FGAPI void FGAPIENTRY __glutInitWithExit(int *argcp, char **argv, void (__cdecl *exit
func)(int));
FGAPI int FGAPIENTRY __glutCreateWindowWithExit(const char *title, void (__cdecl *e
xitfunc)(int));
FGAPI int FGAPIENTRY __glutCreateMenuWithExit(void (* func)(int), void (__cdecl *exi
tfunc)(int));
#ifndef FREEGLUT_BUILDING_LIB
#if defined(__GNUC__)
#define FGUNUSED __attribute__((unused))
#else
#define FGUNUSED
#endif
static void FGAPIENTRY FGUNUSED glutInit_ATEXIT_HACK(int *argcp, char **argv)
{ __glutInitWithExit(argcp, argv, exit); }
```

```
#define glutInit glutInit_ATEXIT_HACK
static int FGAPIENTRY FGUNUSED glutCreateWindow_ATEXIT_HACK(const char *titl
e) { return __glutCreateWindowWithExit(title, exit); }
#define glutCreateWindow glutCreateWindow_ATEXIT_HACK
static int FGAPIENTRY FGUNUSED glutCreateMenu_ATEXIT_HACK(void (* func)(int)
) { return __glutCreateMenuWithExit(func, exit); }
#define glutCreateMenu glutCreateMenu_ATEXIT_HACK
#endif
#endif

#ifdef __cplusplus
    }
#endif

/*** END OF FILE ***/

#endif /* __FREEGLUT_STD_H__ */
```