NAME

xgl_multi_elliptical_arc – draws a list of 3D elliptical arcs

SYNOPSIS

```
#include <xgl/xgl.h>

void xgl_multi_elliptical_arc (
        Xgl_ctx         ctx,
        Xgl_ell_list    *ell_list );
```

DESCRIPTION

Purpose

This operator draws a list of elliptical arcs according to the graphics state described in the 3D Context *ctx*. Each elliptical arc in the list is transformed by the current transformation matrix (see **XGL_CTX_VIEW_TRANS**(3) and **XGL_CTX_MODEL_TRANS**(3)). The elliptical arcs are rendered in the plane described by the two direction vectors provided with each arc. The result is then projected using the current set of Transforms, so that all of the standard 3D pipeline applies. The elliptical arcs are approximated as polygons, using the following context attributes:

XGL_CTX_NURBS_CURVE_APPROX(3)
XGL_CTX_NURBS_CURVE_APPROX_VAL(3)
XGL_CTX_MAX_TESSELLATION(3)
XGL_CTX_MIN_TESSELLATION(3)

Parameters

ctx The XGL 3D Context containing the graphics state used when processing **xgl_multi_elliptical_arc**(3)**.**

ell_list A pointer to the list of arcs to draw. It is a structure defined as:

```
typedef struct {
        Xgl_usgn32          num_ells;   /* number of elliptical arcs */
        Xgl_multiell_type   type;       /* data type of multiell */
        Xgl_bbox            *bbox;      /* optional box containing all
                                           elliptical arcs, NULL means no box */
        union {                         /* union of pointers to ells */
                Xgl_ell_f3d     *f3d;
                Xgl_ell_af3d    *af3d;
                Xgl_ell_d3d     *d3d;
                Xgl_ell_ad3d    *ad3d;
        } ells;
} Xgl_ell_list;
```

ell_list→num_ells is the number of arcs in the list.

ell_list→type is the data type of the center point and radius used to define each arc in the list. *Xgl_multiell_type* currently defines the following enumerated values as possible multiell data types:

XGL_MULTIELLARC_F3D
XGL_MULTIELLARC_AF3D
XGL_MULTIELLARC_D3D
XGL_MULTIELLARC_AD3D

Ultimately, this determines which member of the union *ell_list→ells* (see below) points to the elliptical arc data.

ell_list→bbox points to a structure describing a bounding box for all of the arcs in the list. This may be NULL to indicate that no bounding box information is available.

ell_list→ells is a union of pointers, one of which points to the array of structures that defines each elliptical arc in the list. The *ell_list→ells* structure for elliptical arcs of type XGL_MULTIELLARC_F3D looks as follows:

```
typedef struct {
        Xgl_pt_flag_f3d   center;           /* center point */
        Xgl_pt_f3d        dir[3];           /* dir. vectors and normal vector */
        Xgl_boolean       dir_normalized;   /* TRUE when dir[] are unit vectors */
        Xgl_boolean       dir_normal;       /* TRUE if dir[2] is a valid vector */
        float             major_axis,       /* major and minor axis */
                          minor_axis;
        float             rot_angle;        /* ellipse rotation angle */
        float             start_angle,      /* start and stop angles */
                          stop_angle;
} Xgl_ell_f3d;
```

See **xgl_struct**(3) for structures defining other elliptical arcs in the list.

center and *major/minor* axis members of the structure are *Xgl_pt_flag_f3d* and *float*, respectively.

The first two elements of the field *dir[3]* contain the two orthogonal vectors in model coordinates (MC) used to define the plane on which the elliptical arc is going to be built. The third element is used to specify a normal vector for the arc.

dir_normalized should be set to TRUE when the vectors are normalized (that is, unit vectors). *dir_normal* should be set to TRUE when the third element of *dir* is set to a valid vector. Otherwise, XGL will compute a normal vector to the elliptical arc, using the two vectors specified for the arc plane.

When specifying type XGL_MULTIELLARC_AF3D for elliptical arcs, XGL generates *annotation* elliptical arcs. In other words, the elliptical arcs XGL generates are guaranteed to be in a plane parallel to the projection plane. In this particular case, the center of each arc represents the reference point and is specified in model coordinates. The major and minor axes are specified in virtual device coordinates (VDC). There is no need to specify any direction vectors, and the primitive is always front-facing.

As shown in the *Xgl_ell_f3d* structure, each elliptical arc is specified by an application-defined center point, major/minor axes, rotation angle, and start and stop angle. The *center* and *major/minor* axes are specified in the application's coordinate system, unless the application chooses to render annotation ellipses, as defined earlier. The *rotation* angle is the angle that specifies the rotation of the ellipse, measured between the axis defined by *dir[0]* in the plane of the ellipse and the major axis.

The *start_angle* is the angle from the axis defined by *dir[0]* in the plane of the ellipse to the start point of the arc. The *stop_angle* is the angle from the axis defined by *dir[0]* in the plane of the ellipse to the end point. All angles are specified in radians.

Elliptical arcs can be displayed in one of three ''styles'' depending on the current setting for the attribute **XGL_CTX_ARC_FILL_STYLE**(3). If the arc is closed (that is, if XGL_CTX_ARC_FILL_STYLE is set to either XGL_ARC_SECTOR or XGL_ARC_CHORD), the following surface fill attributes control the appearance of the enclosed area of each arc:

XGL_CTX_SURF_FRONT_FILL_STYLE(3)
XGL_CTX_SURF_FRONT_FPAT(3)
XGL_CTX_SURF_FRONT_FPAT_POSITION(3)
XGL_CTX_SURF_FRONT_COLOR(3)

The following equivalent attributes for back-facing polygons also apply.

XGL_3D_CTX_SURF_BACK_FILL_STYLE(3)
XGL_3D_CTX_SURF_BACK_FPAT(3)
XGL_3D_CTX_SURF_BACK_FPAT_POSITION(3)
XGL_3D_CTX_SURF_BACK_COLOR(3)

The lighting and shading attributes are also in effect.

Edges can be drawn around closed arcs. To do so, first set the attribute **XGL_CTX_-SURF_EDGE_FLAG**(3) to TRUE. Edges will be drawn around each elliptical arc within the arc list that has the flag associated with its *center* point set. Note that the *center* point of each arc always has a flag associated with it. XGL assumes the application will initialize these flags (to either **0** or **1**) before calling xgl_multi_elliptical_arc. If the flags are not initialized, unpredictable results may occur.

For open arcs (that is, arcs drawn when XGL_CTX_ARC_FILL_STYLE is set to XGL_-ARC_OPEN) the line attributes control how the arc is drawn on the Raster. The following attributes are used:

XGL_CTX_LINE_STYLE(3)
XGL_CTX_LINE_PATTERN(3)
XGL_CTX_LINE_WIDTH_SCALE_FACTOR(3)
XGL_CTX_LINE_CAP(3)
XGL_CTX_LINE_JOIN(3)
XGL_CTX_LINE_MITER_LIMIT(3)
XGL_CTX_LINE_COLOR(3)

XGL_CTX_LINE_ALT_COLOR(3)

For a more complete summary of these attributes, see **xgl_multipolyline**(3)**.**

RETURN VALUE

Nothing is returned.

SEE ALSO

xgl_multipolyline(3)
xgl_object_get(3)
xgl_object_set(3)

XGL_CTX_ARC_FILL_STYLE(3)
XGL_CTX_NURBS_CURVE_APPROX(3)
XGL_CTX_NURBS_CURVE_APPROX_VAL(3)
XGL_CTX_MAX_TESSELLATION(3)
XGL_CTX_MIN_TESSELLATION(3)
XGL_3D_CTX_SURF_BACK_AMBIENT(3)
XGL_3D_CTX_SURF_BACK_COLOR(3)
XGL_3D_CTX_SURF_BACK_COLOR_SELECTOR(3)
XGL_3D_CTX_SURF_BACK_DIFFUSE(3)
XGL_3D_CTX_SURF_BACK_DMAP(3)
XGL_3D_CTX_SURF_BACK_DMAP_NUM(3)
XGL_3D_CTX_SURF_BACK_DMAP_SWITCHES(3)
XGL_3D_CTX_SURF_BACK_FILL_STYLE(3)
XGL_3D_CTX_SURF_BACK_FPAT(3)
XGL_3D_CTX_SURF_BACK_FPAT_POSITION(3)
XGL_3D_CTX_SURF_BACK_ILLUMINATION(3)
XGL_3D_CTX_SURF_BACK_LIGHT_COMPONENT(3)
XGL_3D_CTX_SURF_BACK_SPECULAR(3)
XGL_3D_CTX_SURF_BACK_SPECULAR_COLOR(3)
XGL_3D_CTX_SURF_BACK_SPECULAR_POWER(3)
XGL_3D_CTX_SURF_BACK_TRANSP(3)
XGL_3D_CTX_SURF_DC_OFFSET(3)
XGL_3D_CTX_SURF_FACE_CULL(3)
XGL_3D_CTX_SURF_FACE_DISTINGUISH(3)
XGL_3D_CTX_SURF_FRONT_AMBIENT(3)
XGL_3D_CTX_SURF_FRONT_DIFFUSE(3)
XGL_3D_CTX_SURF_FRONT_DMAP(3)
XGL_3D_CTX_SURF_FRONT_DMAP_NUM(3)
XGL_3D_CTX_SURF_FRONT_DMAP_SWITCHES(3)
XGL_3D_CTX_SURF_FRONT_ILLUMINATION(3)
XGL_3D_CTX_SURF_FRONT_LIGHT_COMPONENT(3)
XGL_3D_CTX_SURF_FRONT_SPECULAR(3)
XGL_3D_CTX_SURF_FRONT_SPECULAR_COLOR(3)
XGL_3D_CTX_SURF_FRONT_SPECULAR_POWER(3)
XGL_3D_CTX_SURF_FRONT_TRANSP(3)
XGL_3D_CTX_SURF_GEOM_NORMAL(3)
XGL_3D_CTX_SURF_NORMAL_FLIP(3)

XGL_3D_CTX_SURF_SILHOUETTE_EDGE_FLAG(3)
XGL_3D_CTX_SURF_TRANSP_BLEND_EQ(3)
XGL_3D_CTX_SURF_TRANSP_METHOD(3)
XGL_CTX_SURF_EDGE_FLAG(3)
XGL_CTX_SURF_FRONT_COLOR(3)
XGL_CTX_SURF_FRONT_COLOR_SELECTOR(3)
XGL_CTX_SURF_FRONT_FILL_STYLE(3)
XGL_CTX_SURF_FRONT_FPAT(3)
XGL_CTX_SURF_FRONT_FPAT_POSITION(3)
XGL_CTX_SURF_INTERIOR_RULE(3)
XGL_CTX_LINE_AA_BLEND_EQ(3)
XGL_CTX_LINE_AA_FILTER_SHAPE(3)
XGL_CTX_LINE_AA_FILTER_WIDTH(3)
XGL_CTX_LINE_ALT_COLOR(3)
XGL_CTX_LINE_CAP(3)
XGL_CTX_LINE_COLOR(3)
XGL_CTX_LINE_COLOR_SELECTOR(3)
XGL_CTX_LINE_JOIN(3)
XGL_CTX_LINE_MITER_LIMIT(3)
XGL_CTX_LINE_PATTERN(3)
XGL_CTX_LINE_STYLE(3)
XGL_CTX_LINE_WIDTH_SCALE_FACTOR(3)

NOTES

xgl_multi_elliptical_arc is available only for 3D Contexts.

The *center* point of each arc must have the flag associated with it set to either **0** or **1** (to turn the arc's edge either Off or On). If the flag is not set by the application, unpredictable results may occur.

If the difference between the *start_angle* and *stop_angle* of an arc is greater than 2π radians, then an arc is drawn instead of an ellipse. For example, if *start_angle* is 0.0 radians and *stop_angle* is $(5*\pi/2)$ radians, then an arc is drawn as if the *stop_angle* were $\pi/2$ radians.