NAME

xgl_nurbs_curve – draws a NURBS curve

SYNOPSIS

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

void xgl_nurbs_curve (
        Xgl_ctx                  ctx,
        Xgl_nurbs_curve          *curve,
        Xgl_bounds_f1d           *range, /* optional */
        Xgl_curve_color_spline   *color_spline ); /* optional */
```

DESCRIPTION

This operator generates a NURBS curve of specified order based on the list of knots in the parameter space, list of control points, and the parametric range. Either rational or non-rational curve is selected, depending on the point type.

Parameters

ctx The XGL 2D and 3D Context containing the graphics state used when processing **xgl_nurbs_curve**(3).

curve A pointer to an *Xgl_nurbs_curve* structure that is defined as follows:

```
typedef struct {
        Xgl_usgn32   order;          /* Order of curve */
        Xgl_usgn32   num_knots;      /* Number of knots */
        float        *knot_vector;   /* Knot vector */
        Xgl_pt_list  ctrl_pts;       /* List of control points */
} Xgl_nurbs_curve;
```

curve→order must be a positive integer.

curve→num_knots is the number of knots and it must be equal to the number of control points plus the order of the spline.

curve→knot_vector is an array of floats which must form a non-decreasing sequence.

curve→ctrl_pts is an array of points. The weight of a rational control point cannot be less than or equal to zero.

range The range parameters define the parametric limits of the displayed curve.

The *Xgl_bounds_f1d* structure is defined as follows:

```
typedef struct {
        float   bmin;
        float   bmax;
} Xgl_bounds_f1d;
```

The minimum value should not be greater than the maximum value, and both values should lie between *curve→knot_vector[order-1]* and *curve→knot_vector[curve→num_knots-order]*. If the range parameter is NULL, the whole curve is displayed.

color_spline

A NURBS curve which describes the color distribution over the curve's geometry. Color spline is supported only in 3D and the parameter is ignored in 2D. It is a pointer to an *Xgl_curve_color_spline* structure that is defined as follows:

```
typedef struct {
        Xgl_usgn32              order;
        Xgl_usgn32              num_knots;
        float                   *knot_vector;
        Xgl_color_homogeneous   *colors;
} Xgl_curve_color_spline;
```

The order, num_knots, and knot_vector elements should meet the same requirement as structure *Xgl_nurbs_curve* above. Color splines need not have the same order, rationality or knot sequence as the geometry spline.

The control points of this color spline are color coordinates. The color coordinates can be in 3D (non-rational) or 3D+w (rational). The values of the color spline control point coordinates may be outside the range normally associated with the color value. The vertex color for any point on the mapped geometry of the curve can be determined by evaluating the color spline at the corresponding parameter value.

The *Xgl_color_homogeneous* union is defined as follows:

```
typedef union {
        Xgl_color_rgbw   rgbw;
} Xgl_color_homogeneous;
```

It is a union of all color types (except index) with w coordinate.

The *Xgl_color_rgbw* structure is defined as follows:

```
typedef struct {
        float   r, g, b;
        float   w;
} Xgl_color_rgbw;
```

If the non-rational color is used, w should be set to 1. The color type has to be consistent with the device color type and should not be indexed.

All attributes that affect line-rendering primitives also apply to this primitive.

The quality of smoothness of the curve can be controlled with the **XGL_CTX_NURBS_-CURVE_APPROX**(3) and **XGL_CTX_NURBS_CURVE_APPROX_VAL**(3) attributes. The former selects the approximation method, and the latter selects the reference measure used in the approximation of the curve. The minimum and the maximum number of segments used to display the curve can be controlled with **XGL_CTX_MIN_TESSELLATION**(3) and **XGL_CTX_-MAX_TESSELLATION**(3) attributes.

RETURN VALUE

Nothing is returned.

SEE ALSO

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

The relationship between supported orders and curve approximation methods for non-rational curves is as follows:

Order	**Supported Approximation Method**
1-15	All methods supported
16-25	XGL_CURVE_CONST_PARAM_SUBDIV_BETWEEN_KNOTS only

The relationship between supported orders and curve approximation methods for rational curves is as follows:

Order	**Supported Approximation Method**
1-8	All methods supported
9-25	XGL_CURVE_CONST_PARAM_SUBDIV_BETWEEN_KNOTS only

Curves of order greater than 25 are not supported. Also, rendering of curves of high order (greater than 6) is likely to be very slow.

XGL currently supports the following point types in xgl_nurbs_curve operator:

Xgl_pt_i2d, Xgl_pt_f2d, Xgl_pt_i2h, Xgl_pt_f2h, Xgl_pt_f3d, Xgl_pt_f3h.