1 OpenGL インターフェース

EusLisp OpenGL インターフェースは、 EusLisp に OpenGL インターフェースを実現し、プラットフォーム非依存の3次元グラフィクス環境を EusLisp 処理系で実現する。

OpenGL アプリケーションインターフェースは、 Silicon Graphics 社製 WS DIGITAL EQUIPMENTS 社製 WS を始め、 Windows-NT、 Windows-95（予定）、さらにサードパーティ[1]のソフトウエア製品の追加により Sun WS (Solaris2)、 PC-UNIX （LINUX 等）の上でも動作する。

また、フリーの OpenGL エミュレーション・ライブラリである、 Mesa GL と併用すると、通常の X 端末にも同等の絵の描画を行なうことができる。

1.1 OpenGL 機能拡張

EusLisp の OpenGL 機能拡張は、 OpenGL ライブラリとのインターフェース、 viewsurface クラスのサブクラスである glviewsurface、 geo:viewer の:draw-one メソッドの書き換え、 gldraw ユーティリティ・ルーチンにより実装される。

glviewsurface [class]

:super **geometry:viewsurface**
:slots

OpenGL 用 viewsurface。 viewer への描画リクエストを OpenGL の描画リクエストに翻訳する。

:create [method]

&key (x 0) (y 0) (size 256) (w size) (h size) (title (string (gensym "GL"))) (displaymode (logior gl:AUX_DOUBLE gl:AUX_RGB gl:AUX_DEPTH)) &allow-other-keys

glviewsurface を作成、初期化する。 **:material** *ambient diffuse specular emission shininess* [method]

現在のマテリアル・カラーを変更する。

:color *color-vector* [method]

現在のカラーを変更する。

:3d-mode [method]

射影変換を透視法射影モードにし、デプス検査、ライティングをイネーブルする。 3次元モデル—シーンの描画に適したモードにする。

:2d-mode [method]

射影変換を正射影にし、デプス検査、ライティングをディスエーブルする。 2次元の描画要素の描画に適したモードにする。

:clear *mask &key (color)* [method]

mask で指定される、フレームバッファの内容を消去する。 color は指定があれば、消去時に用いられる色を設定し直す。

:init *&key &allow-other-keys* [method]

初期化メソッド。

:point *x y* [method]

[1] Template Graphics,.Inc 等

2 次元モードで、点を描く。

:line *x1 y1 x2 y2* [method]

2 次元モードで、直線を描く。

:3d-point *pos* [method]

3 次元モードで点を描く。

:3d-line *start end* [method]

3 次元モードで直線を描く。

:draw-faces *faces* [method]

3 次元モードで面(ポリゴン)を描く。

:rectangle *x y w h* [method]

2 次元モードで矩形を描く。

:draw-point *p* [method]

2 次元モードで、点を描く。

:draw-line *from to* [method]

2 次元モードで、直線を描く。

:draw-rectangle [method]

2 次元モードで矩形を描く。

colormaterial [class]

:super **object**
:slots

モデルのマテリアル指定に用いるデータ構造。

gl:draw-face *face* [function]

ポリゴンの描画単位。凸でホールのない、ポリゴンは glVertex3fv を用いて、そのまま描かれる。そうでないものは GLU ライブラリの tesselation 機能を用いて描かれる。

gl:draw-body *abody* [function]

body の描画単位。

body の描画手順は一度、各面毎の描画要求の列として、その body を描く手順が OpenGL の Display List として定義され、実際の描画はその後、その Display List を呼び出すことにより描画が行なわれる。

Display List の記録は body の plist に attribute :GL-DISPLAYLIST-ID で put される。

linecolor [class]

:super **object**
:slots

点、直線の色指定に用いるデータ構造。

cls [function]

画面をクリアする。

swapb [function]

ダブルバッファ・モードのときのバッファ・スワップを行なう。

resetperspective [function]

現在のviewing から視点、注視点位置を取り出し、 GL のコンテキストに反映させる。

gldraw *&rest things* [function]

視点の再設定 (resetperspective)、フレームバッファのクリア (cls)、描画 (apply #'draw things)、ダブルバッファの交換 (swapb) を順に行ない、シーンの描画を行なう。

things は左から右に順に処理され、その要素が、クラス colormaterial、 linecolor のインスタンスの時は、 GL のカレントコンテキストの色が変更される。すなわち、 (gldraw *color1* *obj1* *color2* *obj2*) では、 *obj1* が *color1* で描かれ、 *obj2* が *color2* で描かれる。

1.2 起動方法

eusgl は OpenGL の補助ライブラリ aux、 glx を用いるのため、環境変数 DISPLAY、 LD_LIBRARY_PATH が適当な値にセットされている必要がある。 aux ライブラリは irix5 ではデフォルトでインストールされていないが、 /usr/share/src/OpenGL 下にソースコードがあるほか、 http://www.sgi.com/Technology/openGL/opengl.html または ftp://sgigate.sgi.com/pub/opengl/ 等から取得できる。

Mesa GL ライブラリを用いる場合は、 LD_LIBRARY_PATH の変更のみで、切替ができる。詳細は /usr/local/eus/gl/src/README.Mesa を参照のこと。 Mesa GL は、 http://www.ssec.wisc.edu/~brianp/Mesa.html または ftp://iris.ssec.wisc.edu/pub/Mesa/ 等から入手できる。

プログラム名 eusgl として起動することで、 OpenGL 拡張機能つきの Euslisp が立ち上がる。

eta3 マニピュレータを描くサンプル /usr/local/eus/gl/src/Sample.l が用意してあり、

```
eusgl$ (load "/usr/local/eus/gl/src/Sample.l")
eusgl$ (load "view")
eusgl$ (gldraw *scene*)
```

等とすると絵が表示できる。