簡易 3D グラフィックス関数解説

あらまし

Java 言語・C 言語から(WebGL から解釈するための) 独自形式の JSON ファイルを作成するための特定用途専用 3D グラフィックスライブラリです。Processing という言語とできるだけ同じ名前で描画関数を用意しています。しかし、いくつかの関数は簡略化されていますし、アニメーションやインタラクション関係の関数はありません。

座標系

初期状態では、画面の中心が原点で、x 軸は右向き、y 軸は下向き、z 軸は手前向きにのびています(要確認)。

#include (C 言語) あるいは import (Java 言語)

Java 版では、

```
import static util.Generic3D.*;
```

としてください。(タートルグラフィックスを使う場合は、変更する必要があります。)

C 版では、

```
#include "web3d.h"
```

としてください。(C 版にはタートルグラフィックス関数は未実装です。)

関数一覧

初期設定・その他

```
void start(void);
```
描画の開始のときに呼び出します。
```
void finish(void);
```
描画の終了のときに呼び出します。

色・属性設定

初期状態は、線なし・塗潰し黒です。

```
void stroke(int color);
```
線の色を設定します。色は 0xRRGGBB の形式で指定します。
```
void strokeWeight(double w);
```
線の太さを設定します。
```
void strokeOpacity(double opacity);
```
線の透明度を 0(完全透明)～ 1(完全不透明)の値で指定します。
```
void noStroke(void);
```
線を描きません。
```
void fill(int color);
```
塗潰しの色を設定します。色は 0xRRGGBB の形式で指定します。
```
void fillOpacity(double opacity);
```
塗潰しの透明度を 0(完全透明)～ 1(完全不透明)の値で指定します。
```
void noFill(void);
```
塗潰ししません。
```
void thickness(double t);
```

二次元図形(文字列など)の厚さを指定します。

三次元基本図形

void sphere(double r);
　　原点に半径 r の球を描きます。
void box(double w, double h, double d);
　　原点に幅 w, 高さ h, 奥行き d の直方体を描きます。
void cone(double r, double h);
　　原点に半径 r, 高さ h の円錐を描きます。
void cylinder(double r, double h);
　　原点に半径 r, 高さ h の円柱を描きます。
void line(double x1, double y1, double z1, double x2, double y2, double z2);
　　(x1, y1, z1) から (x2, y2, z2) へ線分を描きます。
void triangle(double x1, double y1, double z1, double x2, double y2, double z2,
double x3, double y3, double z3);
　　3 つの頂点の座標が (x1, y1, z1), (x2, y2, z2), (x3, y3, z3) の 三角形を描きます。
void point(double x1, double y1, double z1);
　　(x1, y1, z1) に点を描きます。

二次元基本図形

以下の図形には void thickness(double) で指定した厚みがつきます。

void rect(double x, double y, double w, double h);
　　左上の頂点の座標が (x, y, 0)、幅 w、高さ h の長方形を描きます。
void ellipse(double x, double y, double w, double h);
　　中心の座標が (x, y, 0)、幅 w、高さ h の楕円を描きます。
void textFont(char* fontName, double size);
　　文字のフォントとサイズを設定します。(初期値は "monotype", 12です。) いまのところ
　　"monotype" フォントの英数字フォントのみ使用できます。
void text(char* str, double x, double y, double z);
　　文字列 str を座標 (x, y, z) に表示します。

三次元曲線・多角形

void beginShape(void);
　　三次元図形の定義を開始します。
void vertex(double x, double y, double z);
　　beginShape で定義を開始した図形に頂点 (x, y, z)を追加します。
void endShape(boolean close);
　　beginShape で定義を開始した図形の定義を終了します。close が真のときは曲線を閉じます。

注意:

beginShape と endShape の間に vertex 以外の描画関係の関数を呼び出さないようにして下さい。

二次元閉曲線・多角形

以下の図形には thickness で指定した厚みがつきます。

void begin2DShape(void);
　　二次元図形の定義を開始します。
void vertex2D(double x, double y);
　　begin2DShape で定義を開始した図形に頂点 (x, y, 0) を追加します。
void end2DShape();
　　begin2DShape で定義を開始した図形の定義を終了します。図形を閉じ多角形を描きます。

注意:

begin2DShape と end2DShape の間に vertex2D 以外の描画関係の関数を呼び出さないようにして下さい。

画像

void image(char* url, double x, double y, double z);
　　インターネット上の画像を左上の頂点の座標が (x, y, z) の長方形に描画します。
void image(char* url, double x, double y, double z, double w, double h);
　　インターネット上の画像を 左上の頂点の座標が (x, y, z)、幅 w、高さ h の長方形内に描画します。
void object(char* url, double x, double y, double z);
　　インターネット上の 3D 形状(COLLADA 形式)を左上の頂点の座標が (x, y, z) に配置します。

色関連のユーティリティ

色関係のユーティリティは ブックカバー作成用グラフィックス関数と同じです。

乱数その他のユーティリティ

乱数関係のユーティリティは ブックカバー作成用グラフィックス関数と同じです。

座標系変換関係

void translate(double x, double y, double z);
　　以降の描画に使用する座標系を (x, y, z) だけ平行移動します。
void rotateX(double theta);
　　以降の描画に使用する座標系を X 軸を中心に theta(単位ラジアン)回転します。
void rotateY(double theta);
　　以降の描画に使用する座標系を Y 軸を中心に theta(単位ラジアン)回転します。
void rotateZ(double theta);
　　以降の描画に使用する座標系を Z 軸を中心に theta(単位ラジアン)回転します。
void scale(double sx, double sy, double sz);
　　以降の描画に使用する座標系を縦方向に sx, 横方向に sy, 奥行き方向に sz だけ拡大します。
void pushMatrix(void);
　　現在、使用している座標系を保存します。
void popMatrix(void);
　　最近 pushMatrix で保存した座標系を取り出します。
void resetMatrix(void);
　　座標系を初期のものに戻します。

タートルグラフィックス関数(Java版のみ)

以下の関数を利用するときは、

```
import static util.Generic3D.*;
```

の代わりに、

```
import static util.Turtle3D.*;
```

として下さい。

“亀”は最初原点 (0, 0, 0) に ペンを下げた状態で x 軸の向き(右)を向いて、z 軸の負の向きを上にしています(要確認)。

void forward(double len);
　　現在、向いている向きに len だけ移動します。
void backward(double len);
　　現在の向きと逆向きに線を描画せずに len だけ移動します。

void turn(double angle);
　　右方向に angle 度回転します。(左方向に回転するときは負の数を渡します。)
void pich(double angle);
　　angle 度ピッチします。(下向きに angle 度折れます(要確認)。上方向に折れるときは負の数を渡します。)
void bank(double angle);
　　angle 度バンクします。(進行方向を軸として反時計回りに angle 度傾きます(要確認)。時計回りに傾くときは負の数を渡します。)
void penUp(void);
　　ペンを上げます。(この状態で移動しても線を描きません。)
void penDown(void);
　　ペンを下げます。(この状態で移動すると線を描きます。)
void go(double x, double y, double z);
　　現在の位置に関係なく、(x, y, z) に移動します。この間、ペンの状態に関係なく、線は描きません。

表示の仕方

プログラムが生成する JSON を JSON Viewer ページ の [Editor] タブ内の textarea 内にペーストして、セーブし、[Viewer] タブをクリックしてください。

印刷の仕方

最新バージョンの Firefox で印刷します。

- まず、JSON Viewer ページ の [Viewer] タブの画像の上で右クリックし、「画像だけを表示」を選択してください。
- 次に、「ファイル」ー「ページ設定」で「書式」の「印刷方向」を「横」、「余白」をすべて「0」、「ヘッダとフッタ」をすべて「なし」に設定して下さい。
- プリンタの方の設定でも、印刷の向きを「横」に設定して下さい。
- 「ファイル」ー「印刷」で印刷して下さい。

Koji Kagawa