

<http://homepage3.nifty.com/li-chu/OpenGL/OpenGL04.html>

■ 射影

正射影

奥行きが無視されるので、手前の物体も奥の物体も同じ大きさになる。無限遠の望遠レンズともいえる。CAD など、物体の形状確認用に向いている。

透視法射影

視点から遠くに行くほど小さく描画する。3D ゲームなど、奥行きのある画像を作りたいときに向いている。描画角が大きいと、物体が大きくゆがむ。

■ glOrtho

GL.glOrtho(left, right, bottom, top, near, far)

引数はすべて Double 型

正射影で描画範囲を指定する。3 つのコマンドで、唯一 near にマイナスを使える。視点を移動しても、大きさは反映されない (大きさが視点からの距離に依存しないため)。視点を移動する場合、near と far の値は + にしたほうが混乱しにくいかもしれない。

left

描画範囲の左端。視線を 0 として +- で入力。

right

描画範囲の右端。視線を 0 として +- で入力。

bottom

描画範囲の上端。視線を 0 として +- で入力。

top

描画範囲の下端。視線を 0 として +- で入力。

near

視点からどれだけ離れた位置から表示するか。+- で入力。

:far よりも大きな値やマイナスでもエラーにはならないが、混乱しやすいので注意 (far との大小が逆転すると、描画方向も変わる)。

far

視点からどれだけ離れた位置まで表示するか。+- で入力。

■ glFrustum

GL.glFrustum(left, right, bottom, top, znear, zfar)

引数はすべて Double 型

透視法射影で描画範囲を指定する。near と far の値は + のみ。また、near < far となること。上 4 つが同じ値のとき、near が小さいほど広角レンズ、大きいほど望遠レンズで撮影した画像になります。広角レンズになるほど、物体の湾曲が強くなります。また、左図下方のように、near の値だけを変えても、距離だけでなく、描画範囲と画角も大きく変わる。描画範囲を変更する場合は注意が必要 (far のみ場合は問題ない)。

left

描画範囲の左端。視線を 0 として +- で入力。

right

描画範囲の右端。視線を 0 として +- で入力。

bottom

描画範囲の上端。視線を 0 として +- で入力。

top

描画範囲の下端。視線を 0 として +- で入力。

near

視点からどれだけ離れた位置から表示するか。必ず 0 より大きく、かつ far よりも小さな値を入れる。

far

視点からどれだけ離れた位置まで表示するか。必ず 0 より大きく、かつ near よりも大きな値を入れる。

■ 隠面消去

初期化

```
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA | GLUT_DEPTH);
```

iphone の場合

```
glGenRenderbuffersOES(1, &depthRenderbuffer);
glBindRenderbufferOES(GL_RENDERBUFFER_OES, depthRenderbuffer);
glRenderbufferStorageOES(GL_RENDERBUFFER_OES, GL_DEPTH_COMPONENT16_OES, backingWidth,
backingHeight);
glFramebufferRenderbufferOES(GL_FRAMEBUFFER_OES, GL_DEPTH_ATTACHMENT_OES, GL_RENDERBUFFER_OES,
depthRenderbuffer);
```

描画時の処理

```
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
glEnable(GL_DEPTH_TEST);
```