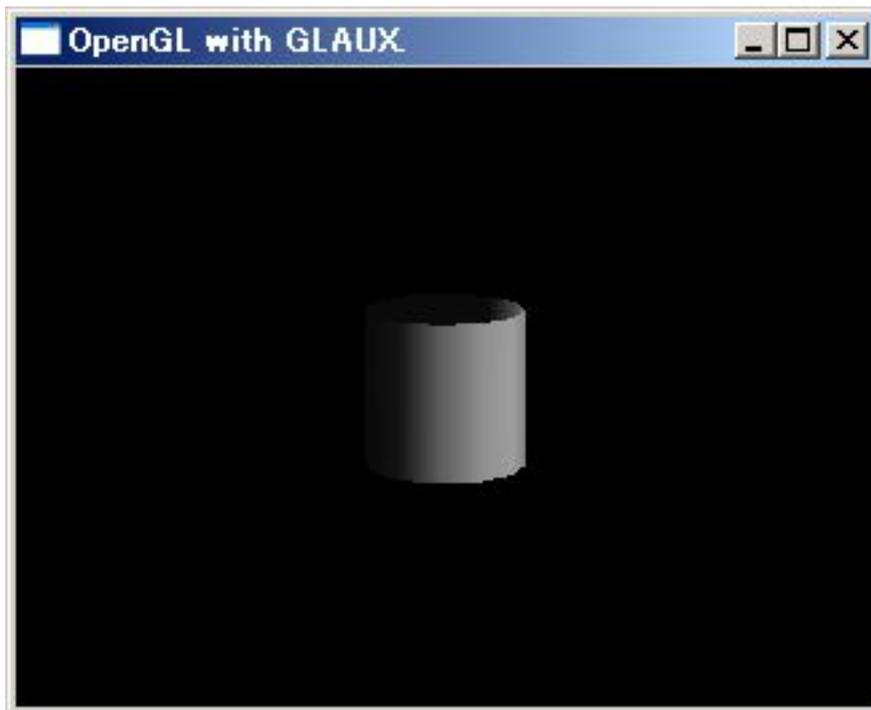


## GLAUX から OpenGL の使用例。



### 必要なファイル

ここから、glaux.lib と glaux.dll を取得。glaux.h が Include/GI ディレクトリにない場合は、ここから取得。glaux.lib は C:\borland\bcc55\Lib\PSDK に配置。glaux.dll は C:\borland\bcc55\Bin に配置。glaux.h は C:\borland\bcc55\Include\GI に配置。

### 参考サイト

[OpenGL Win32 Tutorial OpenGL \(3D グラフィックス\) Visual C Tech Tips](http://64.233.179.104/search?q=cache:wMA0FXfn2UJ:miso.ice.ous.ac.jp/~oka/opengl/section05.html+AUX+DOUBLE&hl=ja&gl=jp&ct=clnk&cd=1&lr=lang_ja&client=firefox)  
[http://64.233.179.104/search?q=cache:wMA0FXfn2UJ:miso.ice.ous.ac.jp/~oka/opengl/section05.html+AUX+DOUBLE&hl=ja&gl=jp&ct=clnk&cd=1&lr=lang\\_ja&client=firefox](http://64.233.179.104/search?q=cache:wMA0FXfn2UJ:miso.ice.ous.ac.jp/~oka/opengl/section05.html+AUX+DOUBLE&hl=ja&gl=jp&ct=clnk&cd=1&lr=lang_ja&client=firefox)

### ソースコード

#### -main.h

```
#include <windows.h>
#include <GL/gl.h>
#include <GL/glu.h>
#include <GL/glaux.h>

#define SCREEN_WIDTH 640
#define SCREEN_HEIGHT 480
#define WINDOW_TITLE "OpenGL with GLAUX."
```

#### -main.cpp

```
#include "main.h"
#include "CGIUtil.h"

int main(int argc, char ** argv) {
    CGIUtil *glUtil = new CGIUtil();
    glUtil->GIMainProc();
    delete glUtil;
    return 0;
}
```

## -CGIUtil.h

```
#include "main.h"

class CGIUtil {
public:
    CGIUtil();
    CGIUtil();
    void GIMainProc(void);
private:
    static GLfloat eye[];

    void GInit(void);
    static void CALLBACK GIDisplay(void);
    static void CALLBACK GReshape(GLint, GLint);
};
```

## -CGIUtil.cpp

```
#include "CGIUtil.h"

GLfloat CGIUtil::eye[] = {40.0, 10.0, 20.0};

CGIUtil::CGIUtil() {
}

CGIUtil::CGIUtil() {
}

void CGIUtil::GIMainProc(void) {
    auxInitDisplayMode(AUX_DOUBLE | AUX_RGBA | AUX_DEPTH);
    auxInitPosition(0, 0, SCREEN_WIDTH, SCREEN_HEIGHT);
    auxInitWindow(WINDOW_TITLE);
    GInit();
    auxReshapeFunc(GReshape);
    auxMainLoop(GIDisplay);
}

void CGIUtil::GInit(void) {
    GLfloat light_position[] = {100.0, 100.0, 100.0, 0.0};

    glLightfv(GL_LIGHT0, GL_POSITION, light_position);

    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    glEnable(GL_DEPTH_TEST);
}

void CALLBACK CGIUtil::GIDisplay(void) {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    gluLookAt(eye[0], eye[1], eye[2], 0.0, 0.0, 0.0, 1.0, 0.0);

    auxSolidCylinder(1.0, 2.0);

    glFlush();
    auxSwapBuffers();
}

void CALLBACK CGIUtil::GReshape(GLint w, GLint h) {
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    gluPerspective(10.0, (GLfloat)w/(GLfloat)h, 1.0, 100.0);

    glMatrixMode(GL_MODELVIEW);
}
```