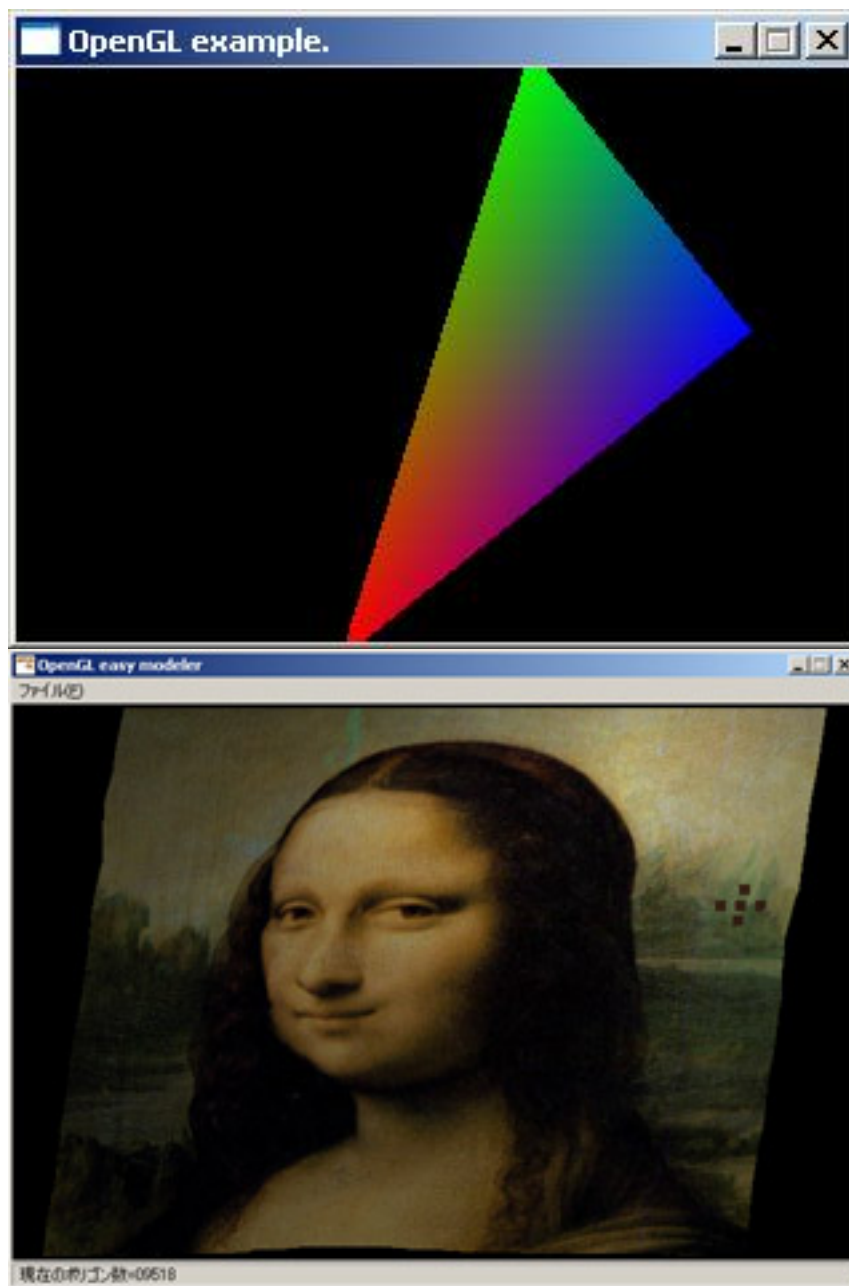
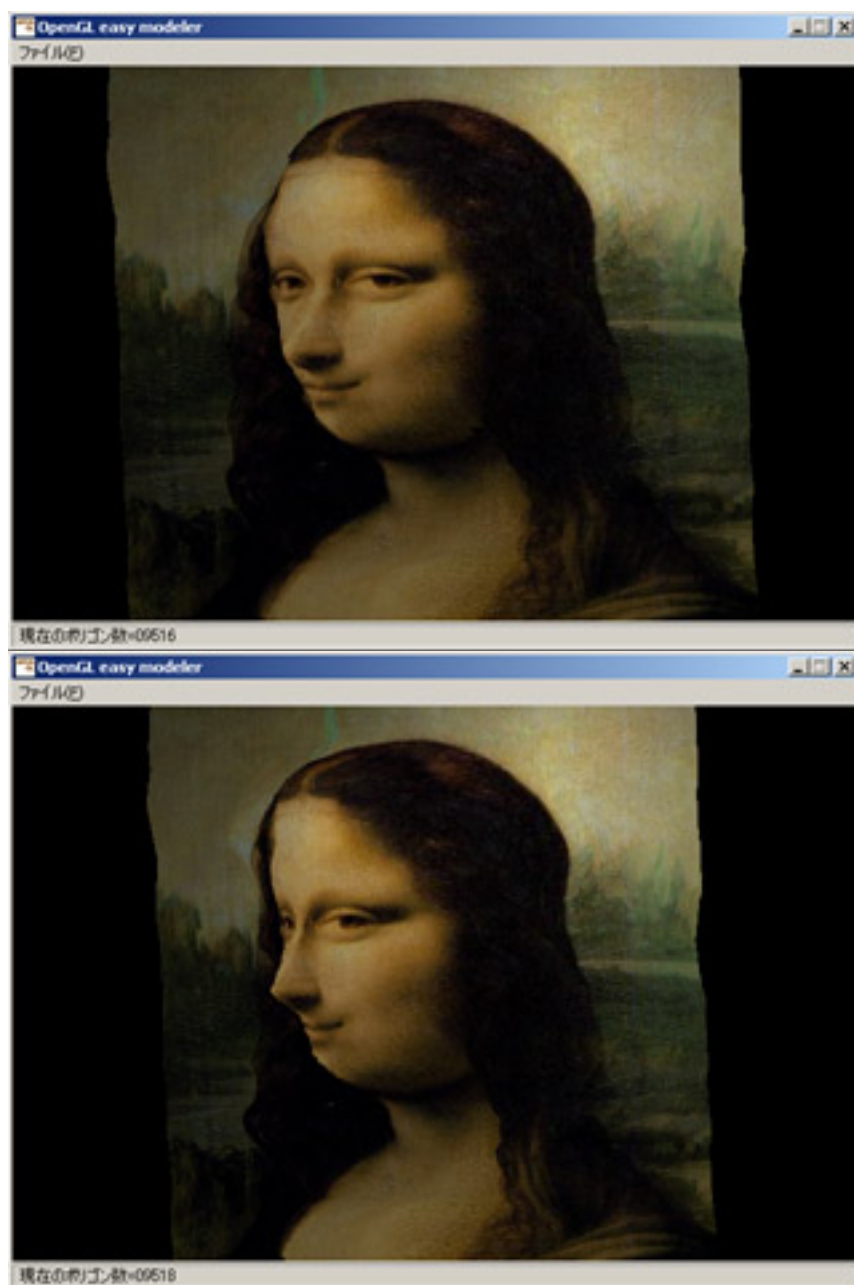
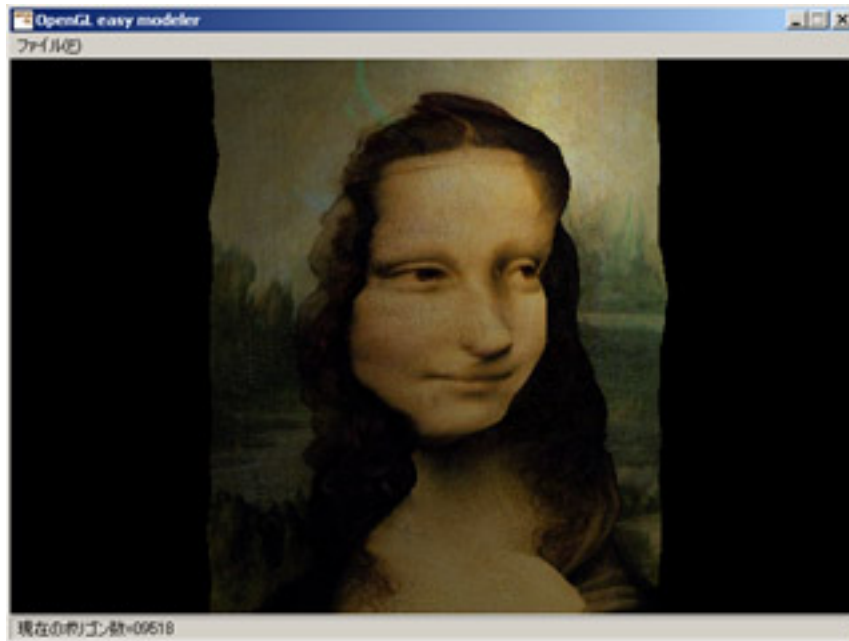


Win32 API から OpenGL の使用例。







-
-

キー操作：

上下左右：回転

- main.cpp

```
#include <windows.h>
#include "util_gl.h"

static HWND hMainWnd;

static LRESULT CALLBACK MainWndProc(HWND hwnd, UINT msg, WPARAM wp, LPARAM lp);

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
    LPSTR lpCmdLine, int nCmdShow)
{
    // テスト用ウィンドウ作成
    WNDCLASS wc;
    ZeroMemory(&wc, sizeof(wc));
    wc.style = CS_HREDRAW | CS_VREDRAW;
    wc.lpfnWndProc = MainWndProc;
    wc.cbClsExtra = 0;
    wc.cbWndExtra = 0;
    wc.hInstance = hInstance;
    wc.hIcon = LoadIcon(NULL, IDI_APPLICATION);
    wc.hCursor = LoadCursor(NULL, IDC_ARROW);
    wc.hbrBackground = (HBRUSH) GetSysColorBrush(COLOR_3DFACE);
    wc.lpszMenuName = NULL;
    wc.lpszClassName = "MAINWND";
    if(!RegisterClass(&wc)) return -1;

    hMainWnd = CreateWindowEx(
        0, "MAINWND", "OpenGL example.",
        WS_OVERLAPPED | WS_SYSMENU | WS_VISIBLE | WS_MINIMIZEBOX,
        CW_USEDEFAULT, CW_USEDEFAULT, 320, 240,
        NULL, NULL, hInstance, NULL);

    ShowWindow(hMainWnd, SW_SHOW);
    UpdateWindow(hMainWnd);
}
```

```

// メッセージループ
MSG msg;
while(GetMessage(&msg , NULL , 0 , 0) > 0) {
    DispatchMessage(&msg);
}
return msg.wParam;
}

static LRESULT CALLBACK MainWndProc(HWND hwnd, UINT msg, WPARAM wp, LPARAM lp)
{
    static PAINTSTRUCT ps;
    static float ang1 = 45.0f;
    static int timerId;

    switch(msg){
    case WM_CREATE:
    {
        // タイマーセット
        timerId = SetTimer(hwnd, 1, 1, NULL);
    }
    return 0;
    case WM_TIMER:
    {
        ang1 += 5.0f;
        if(ang1 > 360.0f) ang1 -= 360.0f;
        InvalidateRect(hwnd, NULL, TRUE);
    }
    return 0;
    case WM_PAINT:
    {
        // 無効リージョンをクリア (BeginPaint() を使用しないため)
        ValidateRect(hwnd , NULL);
        // OpenGL 描画用の HDC を取得
        HDC hdc = GetGDI( hwnd);
        // ポリゴン描画
        glRotatef(ang1, 1.0, 1.0, 1.0);
        glClear(GL_COLOR_BUFFER_BIT);
        glBegin(GL_POLYGON);
        glColor3f(1.0, 0.0, 0.0); glVertex2f(-0.8, -0.8);
        glColor3f(0.0, 1.0, 0.0); glVertex2f( 0.8, 0.8);
        glColor3f(0.0, 0.0, 1.0); glVertex2f( 0.8, -0.8);
        glEnd();
        glFlush();
        // ダブルバッファ切替
        SwapBuffers(hdc);
        // OpenGL 描画用の HDC を解放
        ReleaseGDI( hwnd, hdc);
    }
    return 0;
    case WM_CLOSE:
    {
        // タイマー解除
        KillTimer(hwnd, timerId);
        PostQuitMessage(0);
    }
    return 0;
    }
    return DefWindowProc(hwnd , msg , wp , lp);
}

```

- util_gl.h

```

#ifndef _UTIL_GL_
#define _UTIL_GL_

#include <windows.h>
#include <GL/gl.h>

// OpenGL 描画用 HDC を取得
HDC GetGDI(HWND hwnd);

// OpenGL 描画用 HDC の解放
void ReleaseGDI(HWND hwnd, HDC hdc);

// OpenGL 終了処理
void ExitGL();

#endif

```

- util_gl.cpp

```
#include "util_gl.h"

static HGLRC hGLRC = NULL;

// OpenGL 描画用 HDC を取得
HDC GetGLDC(HWND hwnd)
{
    HDC hdc;
    int pf;
    static PIXELFORMATDESCRIPTOR pfd = {
        sizeof(PIXELFORMATDESCRIPTOR), 1,
        PFD_DRAW_TO_WINDOW | PFD_SUPPORT_OPENGL | PFD_DOUBLEBUFFER,
        PFD_TYPE_RGBA,
        24, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 32, 0, 0,
        PFD_MAIN_PLANE, 0, 0, 0, 0};

    hdc = GetDC(hwnd);
    pf = ChoosePixelFormat(hdc, &pfd);
    SetPixelFormat(hdc, pf, &pfd);

    if(hGLRC == NULL) hGLRC = wglCreateContext(hdc);
    wglMakeCurrent(hdc, hGLRC);

    return hdc;
}

// OpenGL 描画用 HDC の解放
void ReleaseGLDC(HWND hwnd, HDC hdc)
{
    wglMakeCurrent(hdc, 0);
    ReleaseDC(hwnd, hdc);
}

// OpenGL 終了処理
void ExitGL()
{
    if(hGLRC) wglDeleteContext(hGLRC);
}
```