

Hello Klugier !

I then propose this integration :

```
struct GLLock{
private:
    GLCtrl& ctrl;
    bool swap;
public:
    GLLock(GLCtrl& glCtrl, bool swapBuffer = false);
    ~GLLock();
};
```

```
class GLCtrl : public Ctrl {
    typedef GLCtrl CLASSNAME;
    friend class GLLock;
    /*******
    /**}
```

with for Win32GLCtrl.cpp :

```
GLLock::GLLock(GLCtrl& glCtrl, bool swapBuffer) : ctrl(glCtrl), swap(swapBuffer)
{
    HWND hwnd = ctrl.pane.GetHWND();
    HDC hdc = GetDC(hwnd);
    wglMakeCurrent(hdc, s_openGLContext);
}
```

```
GLLock::~~GLLock()
{
    if(swap){
        HWND hwnd = ctrl.pane.GetHWND();
        HDC hdc = GetDC(hwnd);
        SwapBuffers(hdc);
    }else{
        glFlush();
    }
    wglMakeCurrent(NULL, NULL);
}
```

and for XGLCtrl.cpp :

```
GLLock::GLLock(GLCtrl& glCtrl, bool swapBuffer) : ctrl(glCtrl), swap(swapBuffer)
```

```
{  
    glXMakeCurrent(s_Display, ctrl.win, s_GLXContext);  
}
```

```
GLLock::~~GLLock()  
{  
    if(swap){  
        glXSwapBuffers(s_Display, win);  
    }else{  
        glFlush();  
    }  
    glXMakeCurrent(s_Display, None, NULL);  
}
```

Here is diff file :

## File Attachments

---

- 1) [GLCtrl.h.patch](#), downloaded 247 times
  - 2) [Win32GLCtrl.cpp.patch](#), downloaded 259 times
  - 3) [XGLCtrl.cpp.patch](#), downloaded 262 times
-