Subject: [Proposal] Adding a GLLock struct to GLCtrl
Posted by Xemuth on Wed, 02 Sep 2020 16:55:14 GMT
View Forum Message <> Reply to Message

Hello, with the new version of Glew I moved my application to GLCtrl.
However, in many part of my application, I need to get the OpenGL context to perform many actions. The actual way of executing something in the good OpenGL Context is to call ExecuteGL(Event<> paint, bool swap_buffers). Which force me to surround my code arround this function. It work but not convenient and for some part it block me and force me to change my architecture.

To allow more flexibility in the way we work with GLCtrl I propose to add a structure to GLCtrl named GLLock, which can be invoqued to get the OpenGL context and release it when the object got destroyed :

```
//class GLCtrl : public Ctrl{
     public:
 struct GLLock{
  private:
   HWND hwnd;
   HDC hDC;
  public:
   GLLock(GLCtrl& ctrl);
   ~GLLock();
 };
//};
```

Win32GLCtrl.cpp :

```
GLCtrl::GLLock::GLLock(GLCtrl& ctrl){
 hwnd = ctrl.pane.GetHWND();
 GLCtrl::CreateContext();
 hDC = GetDC(hwnd);
 wglMakeCurrent(hDC, s_openGLContext);
}

GLCtrl::GLLock::~GLLock(){
 wglMakeCurrent(NULL, NULL);
 ReleaseDC(hwnd, hDC);
}
```

XGLCtrl.cpp:

```
GLCtrl::GLLock::GLLock(GLCtrl& ctrl){
 glXMakeCurrent(s_Display, ctrl.win, s_GLXContext);
}
```

```
GLCtrl::GLLock::~GLLock(){
 glXMakeCurrent(s_Display, None, NULL);
}
```

Here is a trivial exemple of how to use it :

```
 bool LoadSTL(){ //This function is used when a button is pressed
  try {
   GLCtrl::GLLock __(canvas); //Canvas is the GLCtrl object
   //Shaders Stuff
   //OpenGL buffer filling and loading
   //Definition of draw
   return true;
  } catch (Exc e) {
   Exclamation(DeQtf(e));
  }
  return false;
 }
```

File Attachments

1) GLCtrl.h.patch, downloaded 287 times
2) Win32GLCtrl.cpp.patch, downloaded 260 times
3) XGLCtrl.cpp.patch, downloaded 260 times