Subject: Re: [Proposal] Adding a GLLock struct to GLCtrl
Posted by mirek on Thu, 03 Sep 2020 07:51:18 GMT
View Forum Message <> Reply to Message

Xemuth wrote on Wed, 02 September 2020 18:55Hello, 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.

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;
 }
```

I am not against, but in the example you have posted this does not really feel like improvement...

```
 bool LoadSTL(){ //This function is used when a button is pressed
  try {
                        canvas.ExecuteGL([&] {
    //OpenGL buffer filling and loading
    //Definition of draw
                        });
   return true;
  } catch (Exc e) {
   Exclamation(DeQtf(e));
  }
  return false;
 }
```