**Comprovació d'errors amb OpenGL**

La forma de comprovar si una crida OpenGL ha provocat un error és cridant la funció [**glGetError**](http://www.opengl.org/sdk/docs/man/xhtml/glGetError.xml). La següent macro us pot ser útil per escriure informació addicional com el fitxer .cpp i la línia:

#define glError() { \  
        GLenum err = glGetError(); \  
        while (err != GL\_NO\_ERROR) { \  
            printf("glError: %s caught at %s:%u\n", (char\*)gluErrorString(err), \_\_FILE\_\_, \_\_LINE\_\_); \  
            err = glGetError(); \  
        } \  
    }

Vigileu perquè si glGetError retorna un codi d'error, pot ser degut a qualsevol funció OpenGL cridada des de l'últim cop que vam cridar a glGetError; no necessàriament correspon a la darrera funció OpenGL cridada.

Aquí teniu un extracte del manual de glGetError:

*glGetError returns the value of the error flag. Each detectable error is assigned a numeric code and symbolic name.* ***When an error occurs, the error flag is set to the appropriate error code value. No other errors are recorded until glGetError is called****, the error code is returned, and the flag is reset to GL\_NO\_ERROR. If a call to glGetError returns GL\_NO\_ERROR, there has been no detectable error since the last call to glGetError, or since the GL was initialized.*

*To allow for distributed implementations, there may be several error flags. If any single error flag has recorded an error, the value of that flag is returned and that flag is reset to GL\_NO\_ERROR when glGetError is called. If more than one flag has recorded an error, glGetError returns and clears an arbitrary error flag value. Thus,* ***glGetError should always be called in a loop****, until it returns GL\_NO\_ERROR,* ***if all error flags are to be reset.***

**Comprovació d'errors al compilar shaders**

En el cas dels shaders, la comprovació es basa en les funcions [**glGetShader**](http://www.opengl.org/sdk/docs/man/xhtml/glGetShader.xml)\* i [**glGetShaderInfoLog**](http://www.opengl.org/sdk/docs/man/xhtml/glGetShaderInfoLog.xml).

Un exemple d’ús és aquest:

void printShaderInfoLog**(**GLint shader**)** **{**

int length**;**

glGetShaderiv**(**shader**,** GL\_INFO\_LOG\_LENGTH**,** **&**length**);**

**if** **(**length **>** 0**)** **{**

GLchar log**[**length**];**

glGetShaderInfoLog**(**shader**,** length**,** **NULL,** log**);**

cout **<<** "Log: " **<<** endl **<<** log **<<** endl**;**

**}**

**}**

…

glCompileShader**(**vertexShaderId**);**

GLint compiled**;**

glGetShaderiv**(**vertexShaderId**,** GL\_COMPILE\_STATUS**,** **&**compiled**);**

**if(**compiled**!=**GL\_TRUE**)** **{**

cout **<<** "Vertex shader error" **<<** endl**;**

printShaderInfoLog**(**vertexShaderId**);**

**}**

…

glLinkProgram**(**programID**);**

GLint linked**;**

glGetProgramiv**(**programID**,** GL\_LINK\_STATUS**,** **&**linked**);**

**if(**linked**!=**GL\_TRUE**)** **{**

cout **<<** "Failed to link shader." **<<** endl**;**

printShaderInfoLog**(**vertexShaderId**);**

**}**