OpenGL EXT
(OpenGL EXTensions)

OpenGL extensions

- New functionality added to OpenGL
  - At first vendor specific (more EXTs for same func possible)

    HP_, NV_, ATI_, SGI_, INTEL_ …
  - If vendors agree on common implementation

    EXT_
  - After approval from OpenGL Architecture Review Board

    ARB_
  - If there is a lot of demand of function, it becomes part of the standard
- Get text list (single long string, space separated)

  str = glGetString(GL_EXTENSIONS)
- Test ext. presence

  isExtensionSupported("GL_EXT_bgra")
  glfwExtensionSupported("GL_EXT_bgra")
  glewIsSupported("GL_EXT_bgra")

Enabling extensions

- In Windows you can directly access only OpenGL up to 1.2 incl.
- Newer functions are present in drivers, but you need to enable it (register entry point)
- Example:

```
hasPointParams = isExtensionSupported("GL_EXT_point_parameters");
if (hasPointParams) {
    glPointParameterfEXT = (PFNGLPOINTPARAMETERFEXTPROC);
    wglGetProcAddress("glPointParameterfEXT");
}
```

- Later you can use standard function name

```
if (hasPointParams) {
    static GLfloat quadratic[3] = { 0.25, 0.0, 1/60.0 };
    glPointParameterfvEXT(GL_DISTANCE_ATTENUATION_EXT, quadratic);
}
```

GLEW

- OpenGL Extension Wrangler
  - Simple library for extensions enabling
  - Register all available extensionns, constants, …

- Usage (w/o error check):

```
#include „glew.h“
#include „wglew.h“
main ()
{
   glewInit();
   wglewInit();
}
```

http://glew.sourceforge.net

Usage (with error check)

⚠ Do not forget: set Visual Studio project directories!

```
// OpenGL Extension Wrangler
#include <GL/glew.h>
#include <GL/wglew.h> //WGLEW = Windows GL Extension Wrangler (change for different platform)

void init_glew(void)
{
    //
    // Initialize all valid GL extensions with GLEW.
    // Usable AFTER creating GL context!
    //
    {
        GLenum glew_ret;
        glew_ret = glewInit();
        if (glew_ret != GLEW_OK)
        {
            std::cerr << "WGLEW failed with error: " << glewGetErrorString(glew_ret) << std::endl;
            exit(EXIT_FAILURE);
        }
        else
        {
            std::cout << "GLEW successfully initialized to version: " << glewGetString(GLEW_VERSION) << std::endl;
        }

        // Platform specific. (Change to GLXEW or ELGEW if necessary.)
        glew_ret = wglewInit();
        if (glew_ret != GLEW_OK)
        {
            std::cerr << "WGLEW failed with error: " << glewGetErrorString(glew_ret) << std::endl;
            exit(EXIT_FAILURE);
        }
        else
        {
            std::cout << "WGLEW successfully initialized platform specific functions." << std::endl;
        }
    }
}
```