6 - Introduction to GLSL & OpenGL extensions

Vertex shader

```
                       version statement, needed for compilation
                         ↙
#version 330 core
                                   vertex attribute, different for all vertices
                                    ↙
layout(location = 0) in vec3 pos;
layout(location = 1) in vec4 clr;

out vec3 vColor; ← connects to in in fragment

uniform mat4 mvp; ← uniform, same for all vertices

void main()
{
                         ← main job transforming to clip space
   gl_Position = mvp * vec4(pos, 1);
   vColor = clr.xyz;
}
```

Fragment shader

```
#version 330 core

layout(location=0) out vec4 color; ← Defines output

in vec3 vColor; ← out in vertex connects to this

uniform float alpha;

void main()
{
   color = vec4(vColor, alpha);
}
```

Uniform variables

```
GLint location = glGetUniformLocation(program, "alpha");
glUseProgram(program);
glUniform1f(location, 0,5f);
            1-dimension float
```

OpenGL extensions

- OpenGL version 1,1

  ```
  #include <GL/gl.h>
  ```

- Newer OpenGL versions

  ```
  #include <glew.h>
  glewInit();
  ```

  } Hides a lot of complexity of initializing extensions features for different graphics cards, but initializes them all.

From OpenGL 3.3, GLSL got on version 3,30 to make it easier, but not backwards compatible with previous versions.