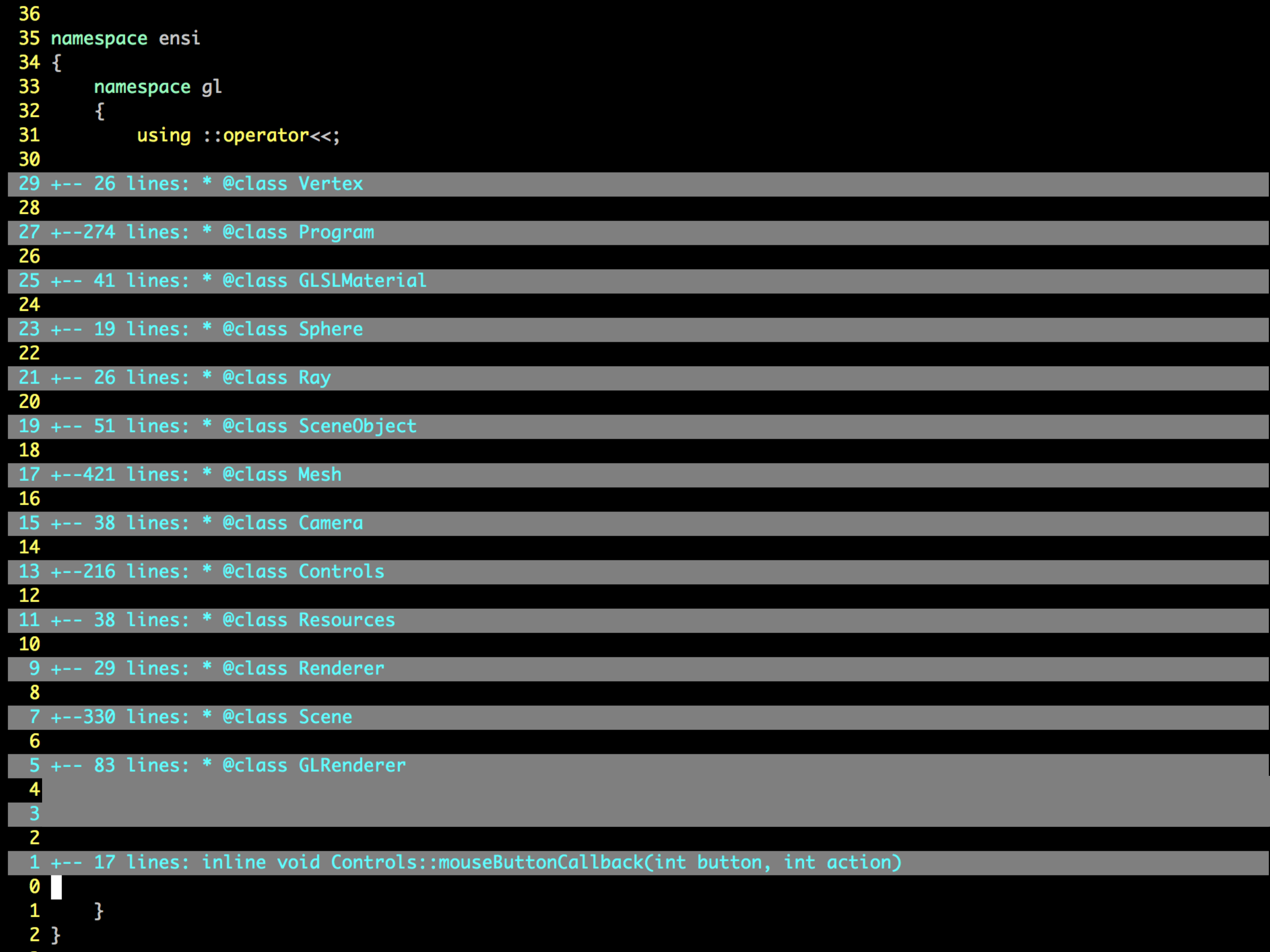
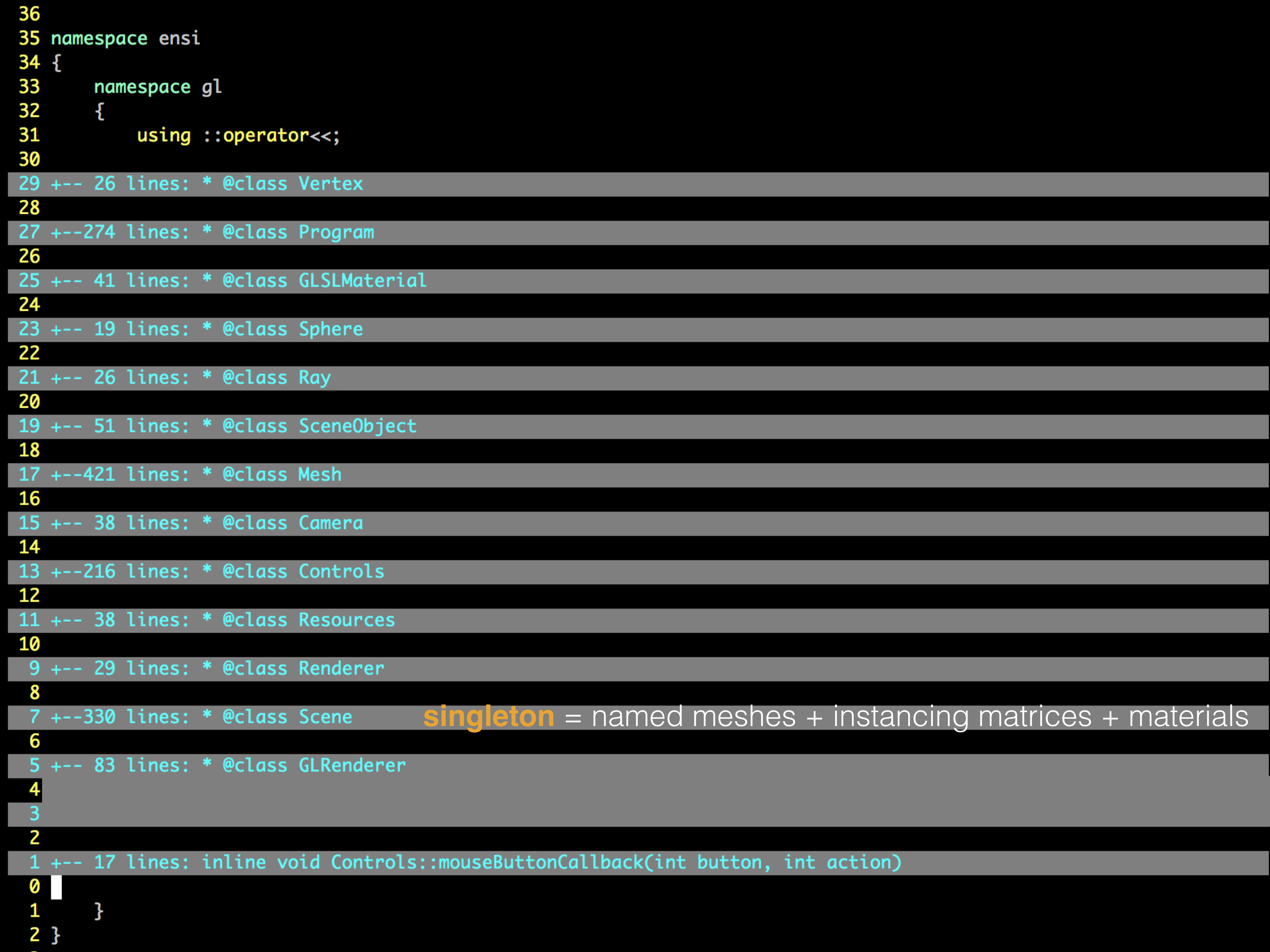
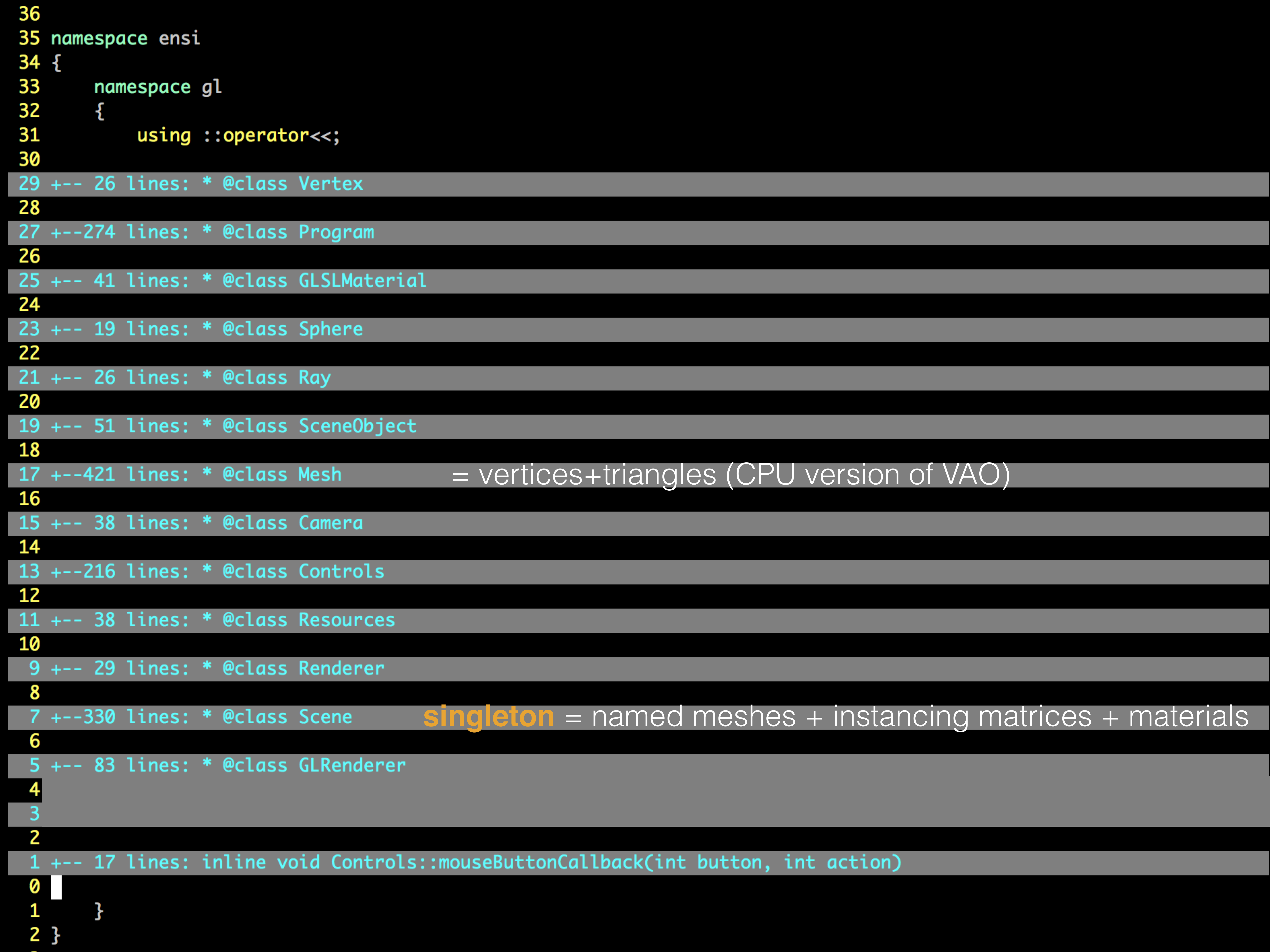


```

namespace ensi
{
    namespace gl
    {
        using ::operator<<;

+-- 26 lines: * @class Vertex

+--274 lines: * @class Program

+-- 41 lines: * @class GLSLMaterial

+-- 19 lines: * @class Sphere

+-- 26 lines: * @class Ray

+-- 51 lines: * @class SceneObject

+--421 lines: * @class Mesh

+-- 38 lines: * @class Camera

+--216 lines: * @class Controls

+-- 38 lines: * @class Resources

+-- 29 lines: * @class Renderer

+--330 lines: * @class Scene

+-- 83 lines: * @class GLRenderer



+-- 17 lines: inline void Controls::mouseButtonCallback(int button, int action)

    }
}
```



```
namespace ensi
{
    namespace gl
    {
        using ::operator<<;

+-- 26 lines: * @class Vertex

+--274 lines: * @class Program

+-- 41 lines: * @class GLSLMaterial

+-- 19 lines: * @class Sphere

+-- 26 lines: * @class Ray

+-- 51 lines: * @class SceneObject

+--421 lines: * @class Mesh

+-- 38 lines: * @class Camera

+--216 lines: * @class Controls

+-- 38 lines: * @class Resources

+-- 29 lines: * @class Renderer

+--330 lines: * @class Scene

+-- 83 lines: * @class GLRenderer



+-- 17 lines: inline void Controls::mouseButtonCallback(int button, int action)

    }
}
```

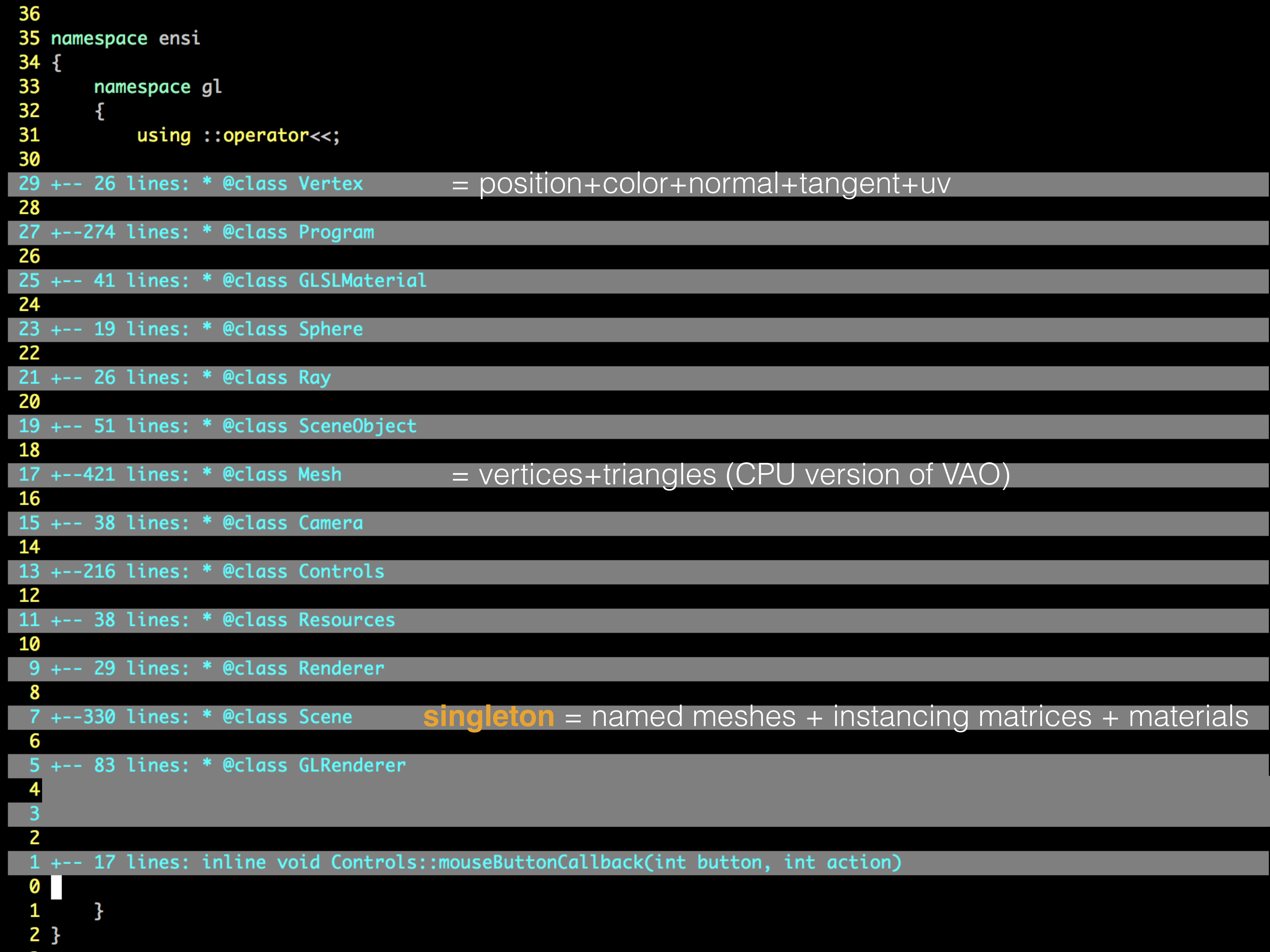
singleton = named meshes + instancing matrices + materials



```
namespace ensi
{
    namespace gl
    {
        using ::operator<<;

+-- 26 lines: * @class Vertex

+--274 lines: * @class Program

+-- 41 lines: * @class GLSLMaterial

+-- 19 lines: * @class Sphere

+-- 26 lines: * @class Ray

+-- 51 lines: * @class SceneObject

+--421 lines: * @class Mesh

+-- 38 lines: * @class Camera

+--216 lines: * @class Controls

+-- 38 lines: * @class Resources

+-- 29 lines: * @class Renderer

+--330 lines: * @class Scene

+-- 83 lines: * @class GLRenderer



+-- 17 lines: inline void Controls::mouseButtonCallback(int button, int action)

    }
}
```

Mesh = vertices+triangles (CPU version of VAO)

Scene **singleton** = named meshes + instancing matrices + materials

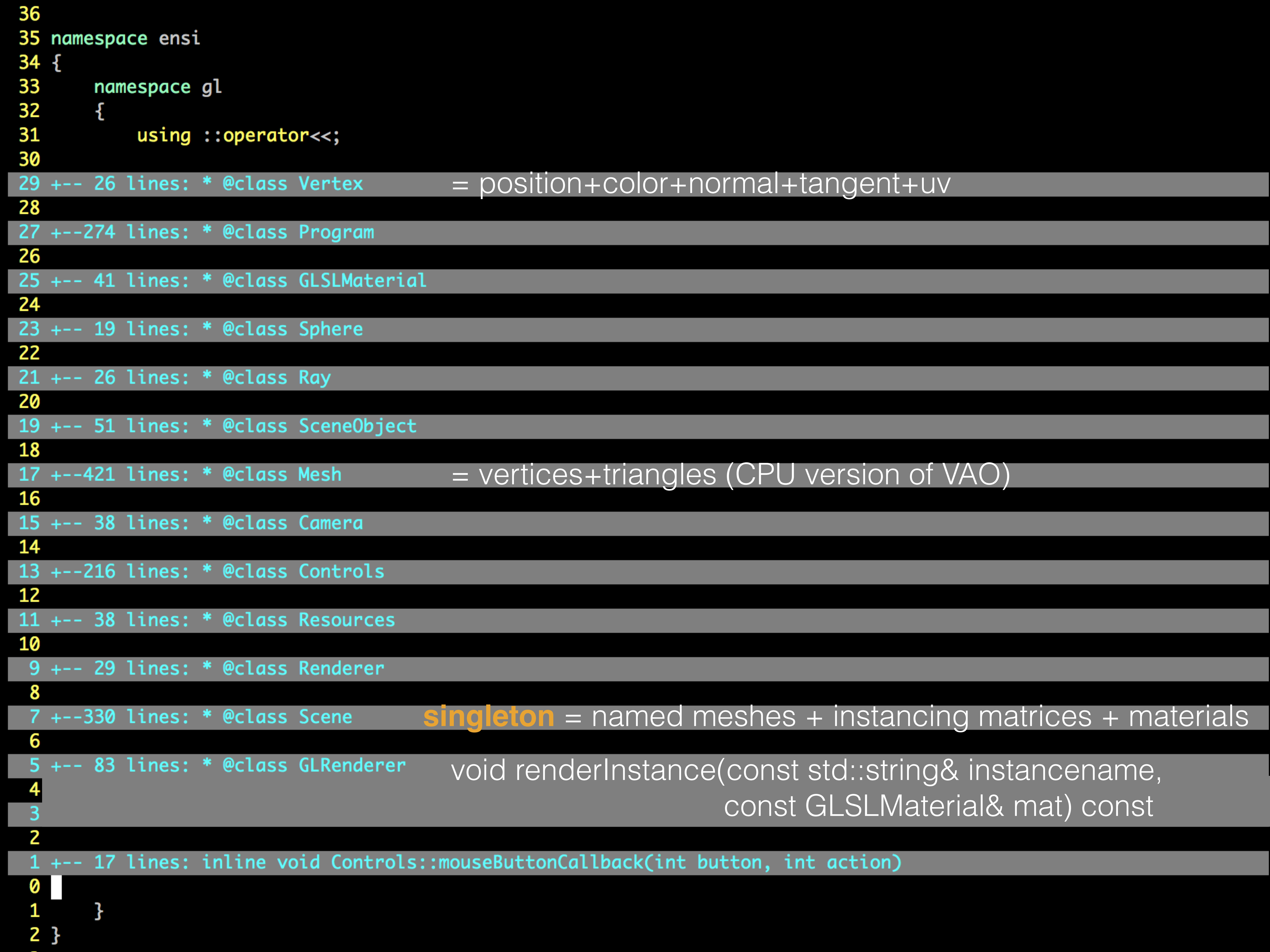


```
namespace ensi
{
    namespace gl
    {
        using ::operator<<;

+-- 26 lines: * @class Vertex

+--274 lines: * @class Program

+-- 41 lines: * @class GLSLMaterial

+-- 19 lines: * @class Sphere

+-- 26 lines: * @class Ray

+-- 51 lines: * @class SceneObject

+--421 lines: * @class Mesh

+-- 38 lines: * @class Camera

+--216 lines: * @class Controls

+-- 38 lines: * @class Resources

+-- 29 lines: * @class Renderer

+--330 lines: * @class Scene

+-- 83 lines: * @class GLRenderer



+-- 17 lines: inline void Controls::mouseButtonCallback(int button, int action)

    }
}
```

= position+color+normal+tangent+uv

= vertices+triangles (CPU version of VAO)

singleton = named meshes + instancing matrices + materials



```
namespace ensi
{
    namespace gl
    {
        using ::operator<<;

+-- 26 lines: * @class Vertex

+--274 lines: * @class Program

+-- 41 lines: * @class GLSLMaterial

+-- 19 lines: * @class Sphere

+-- 26 lines: * @class Ray

+-- 51 lines: * @class SceneObject

+--421 lines: * @class Mesh

+-- 38 lines: * @class Camera

+--216 lines: * @class Controls

+-- 38 lines: * @class Resources

+-- 29 lines: * @class Renderer

+--330 lines: * @class Scene

+-- 83 lines: * @class GLRenderer



+-- 17 lines: inline void Controls::mouseButtonCallback(int button, int action)

    }
}
```

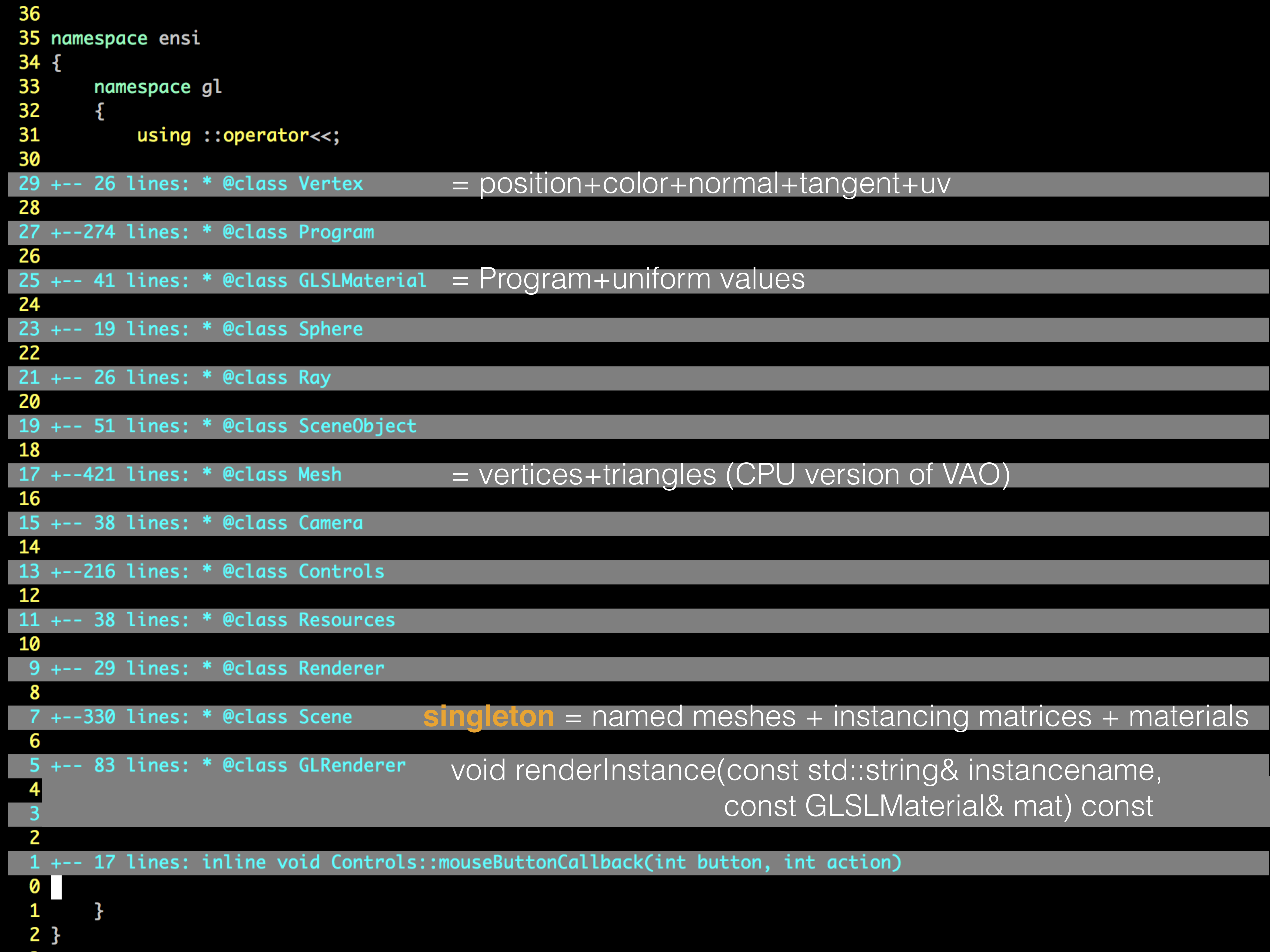
Vertex = position+color+normal+tangent+uv

Mesh = vertices+triangles (CPU version of VAO)

Scene: **singleton** = named meshes + instancing matrices + materials

GLRenderer:

```
void renderInstance(const std::string& instancename,
                    const GLSLMaterial& mat) const
```



```
namespace ensi
{
    namespace gl
    {
        using ::operator<<;

+-- 26 lines: * @class Vertex

+--274 lines: * @class Program

+-- 41 lines: * @class GLSLMaterial

+-- 19 lines: * @class Sphere

+-- 26 lines: * @class Ray

+-- 51 lines: * @class SceneObject

+--421 lines: * @class Mesh

+-- 38 lines: * @class Camera

+--216 lines: * @class Controls

+-- 38 lines: * @class Resources

+-- 29 lines: * @class Renderer

+--330 lines: * @class Scene

+-- 83 lines: * @class GLRenderer



+-- 17 lines: inline void Controls::mouseButtonCallback(int button, int action)

    }
}
```

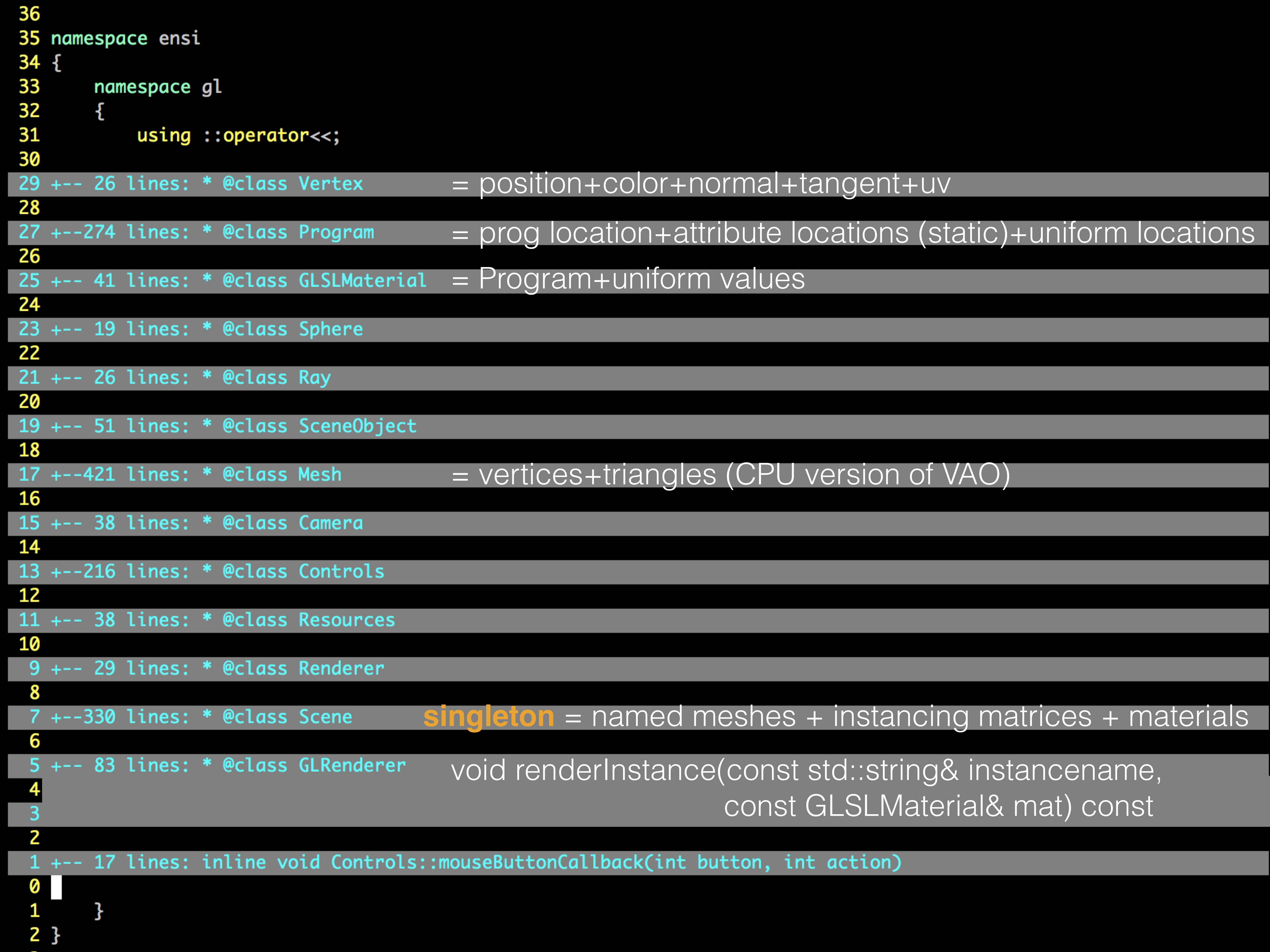
= position+color+normal+tangent+uv

= Program+uniform values

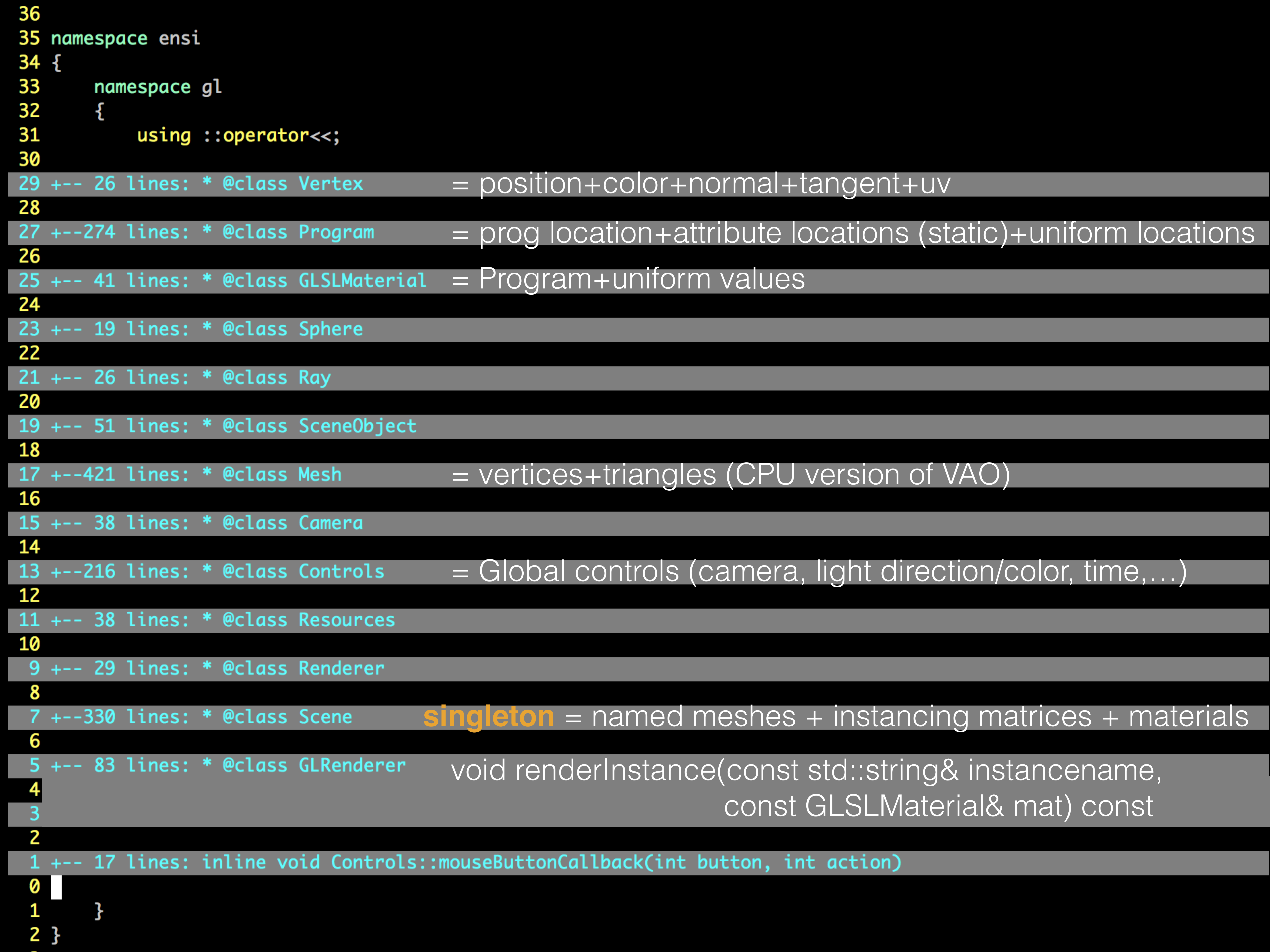
= vertices+triangles (CPU version of VAO)

singleton = named meshes + instancing matrices + materials

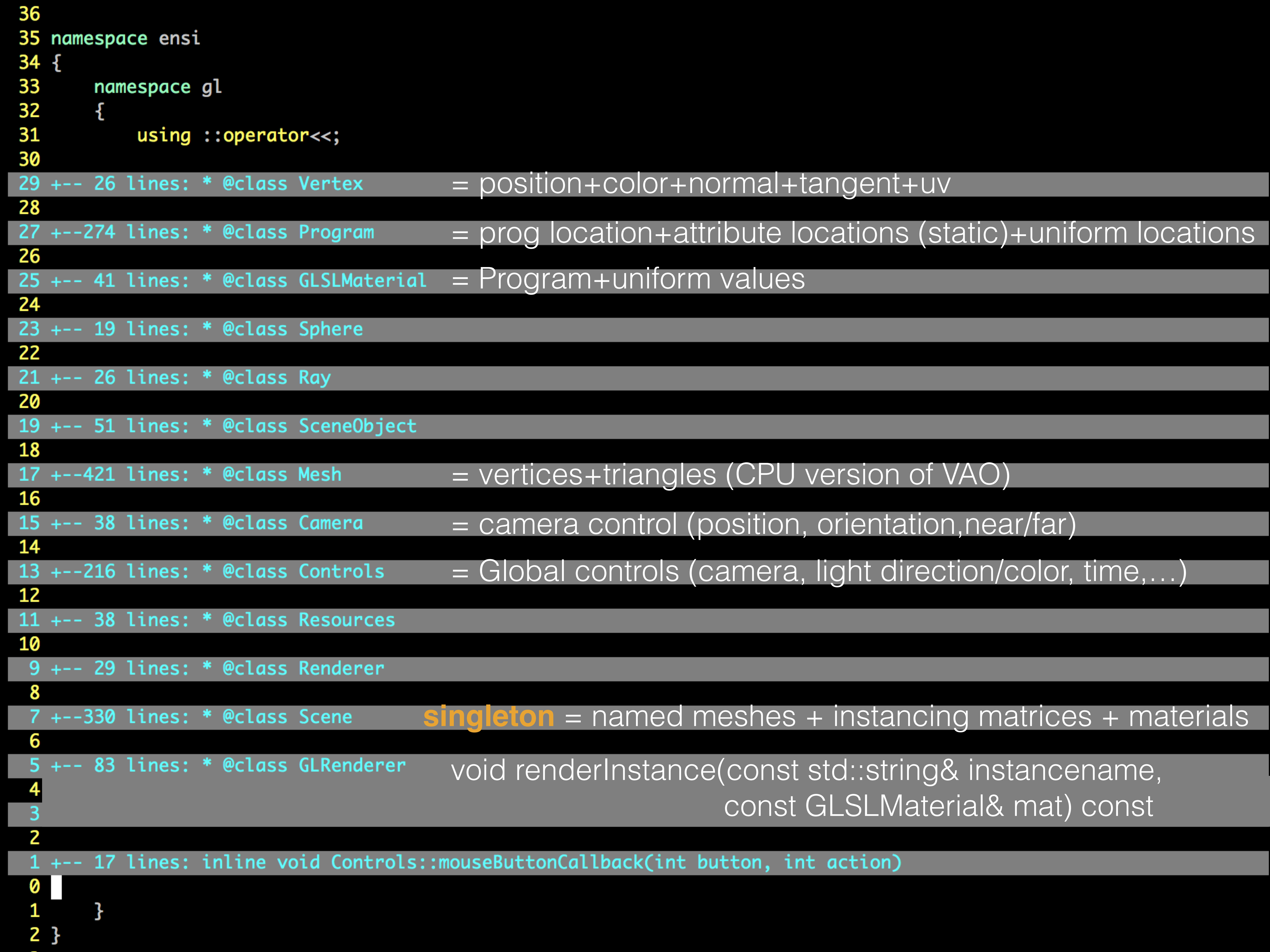
void renderInstance(const std::string& instancename,
const GLSLMaterial& mat) const



```
namespace ensi
{
    namespace gl
    {
        using ::operator<<;

+-- 26 lines: * @class Vertex              = position+color+normal+tangent+uv

+--274 lines: * @class Program             = prog location+attribute locations (static)+uniform locations

+-- 41 lines: * @class GLSLMaterial        = Program+uniform values

+-- 19 lines: * @class Sphere

+-- 26 lines: * @class Ray

+-- 51 lines: * @class SceneObject

+--421 lines: * @class Mesh                = vertices+triangles (CPU version of VAO)

+-- 38 lines: * @class Camera

+--216 lines: * @class Controls

+-- 38 lines: * @class Resources

+-- 29 lines: * @class Renderer

+--330 lines: * @class Scene             singleton = named meshes + instancing matrices + materials

+-- 83 lines: * @class GLRenderer         void renderInstance(const std::string& instancename,
                                                              const GLSLMaterial& mat) const

+-- 17 lines: inline void Controls::mouseButtonCallback(int button, int action)

    }
}
```



```
namespace ensi
{
    namespace gl
    {
        using ::operator<<;

+-- 26 lines: * @class Vertex          = position+color+normal+tangent+uv

+--274 lines: * @class Program         = prog location+attribute locations (static)+uniform locations

+-- 41 lines: * @class GLSLMaterial    = Program+uniform values

+-- 19 lines: * @class Sphere

+-- 26 lines: * @class Ray

+-- 51 lines: * @class SceneObject

+--421 lines: * @class Mesh            = vertices+triangles (CPU version of VAO)

+-- 38 lines: * @class Camera

+--216 lines: * @class Controls        = Global controls (camera, light direction/color, time,…)

+-- 38 lines: * @class Resources

+-- 29 lines: * @class Renderer

+--330 lines: * @class Scene        singleton = named meshes + instancing matrices + materials

+-- 83 lines: * @class GLRenderer    void renderInstance(const std::string& instancename,
                                                          const GLSLMaterial& mat) const


+-- 17 lines: inline void Controls::mouseButtonCallback(int button, int action)

    }
}
```



```

namespace ensi
{
    namespace gl
    {
        using ::operator<<;

+-- 26 lines: * @class Vertex            = position+color+normal+tangent+uv

+--274 lines: * @class Program           = prog location+attribute locations (static)+uniform locations

+-- 41 lines: * @class GLSLMaterial      = Program+uniform values

+-- 19 lines: * @class Sphere

+-- 26 lines: * @class Ray

+-- 51 lines: * @class SceneObject

+--421 lines: * @class Mesh              = vertices+triangles (CPU version of VAO)

+-- 38 lines: * @class Camera            = camera control (position, orientation,near/far)

+--216 lines: * @class Controls          = Global controls (camera, light direction/color, time,…)

+-- 38 lines: * @class Resources

+-- 29 lines: * @class Renderer

+--330 lines: * @class Scene       singleton = named meshes + instancing matrices + materials

+-- 83 lines: * @class GLRenderer      void renderInstance(const std::string& instancename,
                                                    const GLSLMaterial& mat) const

+-- 17 lines: inline void Controls::mouseButtonCallback(int button, int action)

    }
}
```