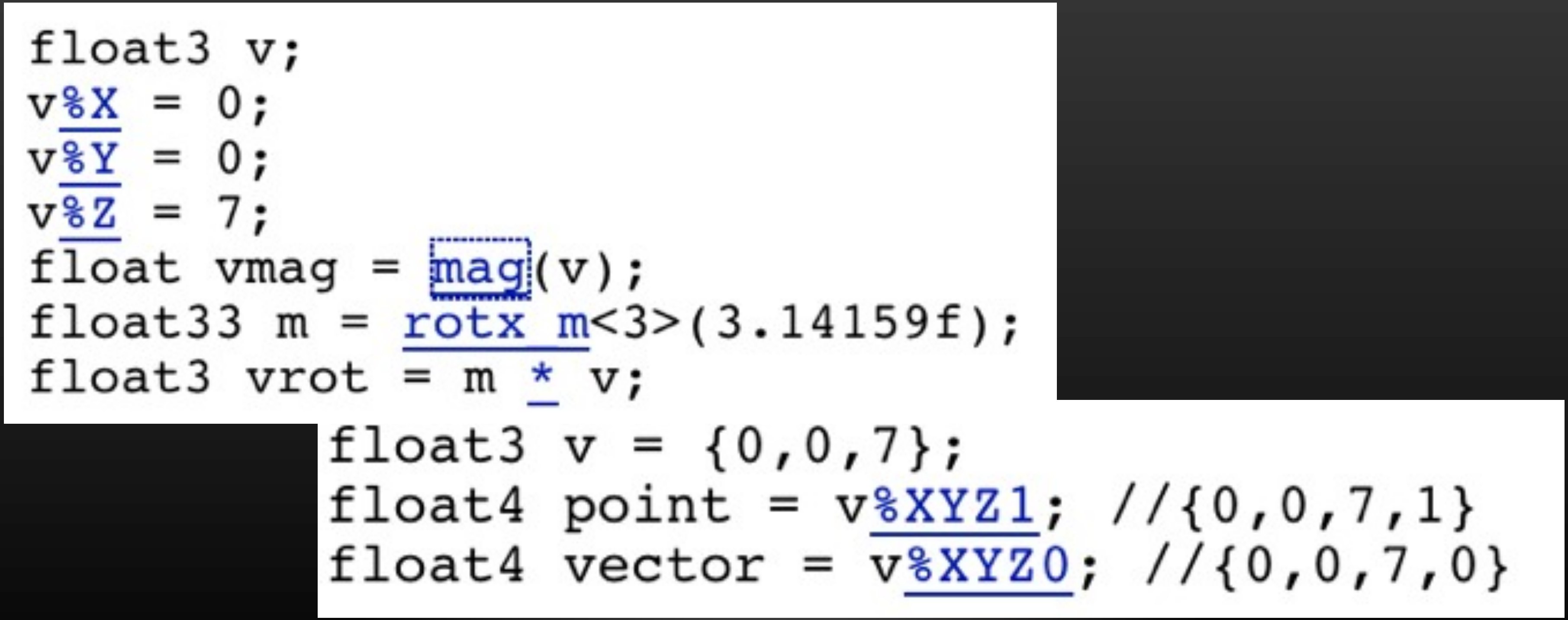
PROPOSED BOOST::QVM

Quaternion, Vector & Matrix
written by Emil Dotchevski
http://www.revergestudios.com/boost-qvm/index.html

overview presentation for BoostCon “Library in a Week” Session
by Fabio Fracassi

WHAT DOES IT DO?

- Basic fixed width Vector, Matrix & Quaternion Math
- Generic – uses type traits and SFINAE to be able to use with user defined Q, V, M types



```
float3 v;
v%X = 0;
v%Y = 0;
v%Z = 7;
float vmag = mag(v);
float33 m = rotx_m<3>(3.14159f);
float3 vrot = m * v;
```

```
float3 v = {0,0,7};
float4 point = v%XYZ1; //{0,0,7,1}
float4 vector = v%XYZ0; //{0,0,7,0}
```

FIRST IMPRESSION

- Small Scope: Basic LA operations for 2d, 3d, 4d.
  - Most Graphics needs should be fullfilled
  - Brings std math operations like sin, cos, …
  - Swizzeling
- Understandable Documentation (Tutorial), reference incomplete
- Uses boost:: exception, enable_if, static_assert
- Superficial code-gazing: Looks straightforward and clean.
  - Except for some code generation I didn’t immediately understand
- Can integrate third-party Q, V, M types (even e.g. `float[3]`)
  - Also brings its own (`qvm::vec<double, 3>`)

WHAT IS MISSING

- Documentation is incomplete
    - How to add more functions
    - Reference
    - Performance
- Missing: How does it compare to, integrate with similar libs in boost (or wide use) like:
    - ublas
    - NT2
    - Eigen

THANK YOU

Have a look at some test code prepared by Jeff Trull at:

http://github.com/jefftrull/QVM-Test