Diagrams

```
                              ┌─────────────────────────┐
                              │     QBaseMatrix4x3       │
                              ├─────────────────────────┤
                              │ +ij[4][3] : FloatType    │
                              ├─────────────────────────┤
                              │                          │
                              └─────────────────────────┘
                                         △
                                         │
                              ┌─────────────────────────┐
                              │       QMatrix4x3         │
                              ├─────────────────────────┤
                              │ +Identity : QMatrix4x3   │
                              │ +ZeroMatrix : QMatrix4x3 │
    ┌──────────────────┐      ├─────────────────────────┤
    │   QTriangle3     │      │ +operator+()             │        ┌──────────────┐
    ├──────────────────┤      │ +operator-()             │        │  «datatype»  │
    │                  │----->│ +operator*()             │        │ V4x32-FLOAT  │
    ├──────────────────┤      │ +operator/()             │------->├──────────────┤
    │                  │      │ +operator+=()            │        │              │
    └──────────────────┘      │ +operator-=()            │        └──────────────┘
                              │ +operator*=()            │
                              │ +operator/=()            │        ┌──────────────┐
                              │ +operator==()            │------->│  «datatype»  │
                              │ +operator!=()            │        │   FloatType  │
                              │ +Reverse()               │------->├──────────────┤
                              │ +Transpose()             │        │              │
                              │ +ResetToIdentity()       │        └──────────────┘
                              │ +ResetToZero()           │
                              │ +IsIdentity()            │
                              │ +IsZero()                │
                              │ +GetDeterminant()        │
                              │ +ToString()              │
                              └─────────────────────────┘
```

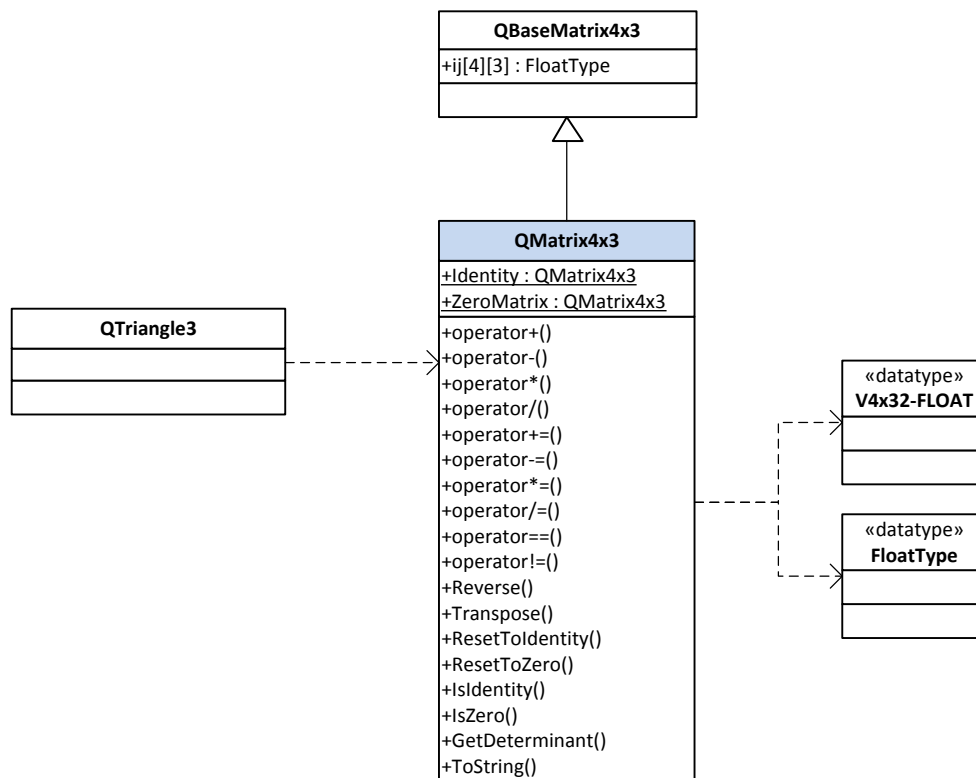## Knowledge Requirements

- Math
- See: Introduction to 3D Game Programming with DirectX 9.0, Part I.
- See: *<Program Files Folder>\Microsoft DirectX SDK (June 2010)\Documentation\DirectX9\windows_graphics.chm*, from DirectX SDK. Search for D3DXMATRIX4 and D3DXMatrix in "Index" tab.
- See: *http://www.zator.com/Cpp/E4_9_18.htm* to refresh operators overloading knowledge.

Functional Specifications

- Override default constructor. Sets attributes to zero.
- Override copy constructor.
- Implement constructor that receives a QBaseMatrix4x3 type.
- Implement constructor that receives 3 FloatTypes, one for each Euler angle. The matrix must be initialized as if it was a rotation matrix, using inputs angles.
- Implement constructor that receives only 1 FloatType. Set all attributes to that value.
- Implement constructor that receives a 12-FloatTypes array. Remember that Quimera Engine uses a row x column convention.
- Implement constructor that receives a pointer-to-FloatType. The pointer should point to a dynamically allocated 12-FloatTypes array. Remember that Quimera Engine uses a row x column convention.
- Implement constructor that receives four V4x32-FLOATs. One per row, ignoring fourth component.
- It is not necessary to override default destructor.
- It is not necessary to override assign operator.
- Operator* must offer an overload that receives a FloatType (product by scalar).

- Operator* must offer an overload that receives a QBaseMatrix4x3.
- Operator/ implements "division by scalar" only, there is no division operation between matrices.
- Operator/= implements "division by scalar" only, there is no division operation between matrices.
- A global operator* must be implemented in order to let a FloatType be multiplied by a QMatrix4x3. It's not the same QMatrix4x3 * FloatType than FloatType * QMatrix4x3.
- A global operator/ must be implemented in order to let a FloatType be multiplied by a QMatrix4x3. It's not the same QMatrix4x3 / FloatType than FloatType / QMatrix4x3.
- ResetToZero sets all matrix's elements to 0.
- ToString format: "M4x3(11, 12, 13)(21, 22, 23)(31, 32, 33) (41, 42, 43)". Use STL string.

Design / Technical Requirements

- Use member initialization lists.
- Remember using "explicit" when constructors receive only one parameter.
- No virtual methods.
- Use by-reference parameters always.
- Operator== and Operator!= must have Epsilon value into account.
- Use FloatType constants to store values like 0.
- All methods should be inline.
- No exceptions.
- No error codes.
- No profiling.
- Check for division by zero. Use asserts.
- Respect diagram names.

Support People

- Thund.