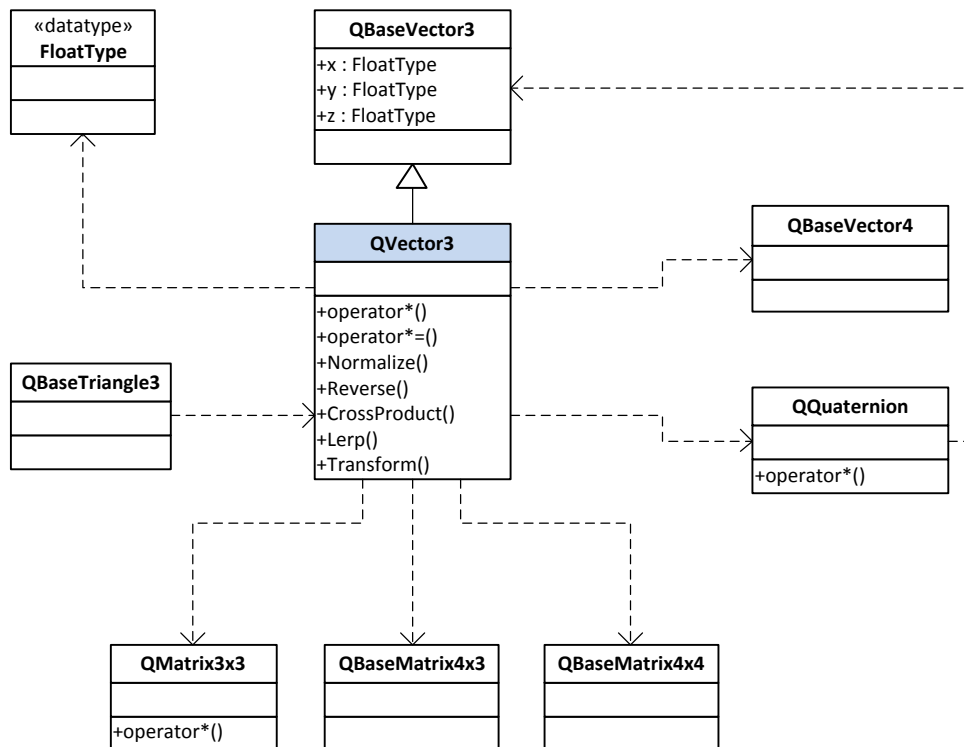


## Diagrams



## 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 D3DXVECTOR3 and D3DXVec3 in "Index" tab.
- See: <http://euclideanspace.com/maths/algebra/matrix/index.htm>.
- See: <http://euclideanspace.com/maths/algebra/realNormedAlgebra/quaternions/index.htm>.
- See: <http://euclideanspace.com/maths/geometry/elements/plane/index.htm>.

## Functional Specifications

- Implement constructor that receives a **QBaseVector4** type.
- **Operator\*** must offer an overload that receives a **QBaseMatrix3x3** (the vector is a 1x3 matrix). Internally, use **QMatrix3x3** functionality.
- **Operator\*=** must offer an overload that receives a **QBaseMatrix3x3** (the vector is a 1x3 matrix). Internally, use **QMatrix3x3** functionality.
- **Operator\*** must offer an overload that receives a **QQuaternion**. Internally, use **QQuaternion** functionality.
- **Operator\*=** must offer an overload that receives a **QQuaternion**. Internally, use **QQuaternion** functionality.
- **Normalize** must offer an overload that receives a **QBaseVector3**. The result is a normalized copy of the vector and the original vector isn't modified. Internally, use the previously defined **Normalize** method.
- **Reverse** must offer an overload that receives an output **QBaseVector3**. The result is a reversed copy of the vector and the original vector isn't modified. Internally, use the previously defined **Reverse** method.

- CrossProduct must offer an overload that receives two QBaseVector3s. The second parameter stores the result, which is assigned to the resultant perpendicular vector. The original vector isn't modified. Internally, use the previously defined CrossProduct method.
- Lerp must offer an overload that receives an output QBaseVector3. The result is a "scaled" copy of the vector and the original vector isn't modified. Internally, use the previously defined Lerp method.
- Transform must offer an overload that receives a QBaseMatrix4x3. Matrix transformations will be applied to the vector.
- Transform must offer an overload that receives a QBaseMatrix4x3 and an output QBaseVector3. Matrix transformations will be applied to the copy of the vector, stored in the output parameter. Internally, use the other overloaded method.
- Transform must offer an overload that receives a QBaseMatrix4x4. Matrix transformations will be applied to the vector.
- Transform must offer an overload that receives a QBaseMatrix4x4 and an output QBaseVector3. Matrix transformations will be applied to the copy of the vector, stored in the output parameter. Internally, use the other overloaded method.

## Design / Technical Requirements

- No virtual methods.
- Use by-reference parameters always.
- Try to avoid square roots.
- All methods should be inline.
- No exceptions.
- No error codes.
- No profiling.
- Respect diagram names.

## Support People

- Thund.

