A03 – Advanced transforms

The WebGL application contained in file `index.html`, wants to perform 4 advanced transformations using the code written in file `move.js`.

If you look at the code in `move.js`, you will see that all transforms are initialized to the identity matrix (which performs no transform). Your goal is to modify such matrices to obtain the desired effect.

If you open `index.html` in Google Chrome or in some other WebGL supported browser, you will see at top of the page the requested transform, and a wireframe view of the wanted result. If you press space, it will add a filled view of the object obtained using the transform you supplied in the file `move.js`. If the wireframe and solid object matches, you have done it right! You can press space and move to the next transform.

In this exercise, you can use if you wish (but you are not required to, if you prefer to perform computation externally), third-party libraries to build basic rotation, scale, translation and shear matrix, as well to perform matrix multiplication. For example, you can use the already included library **utils.js**, that exposes the following functions:

`utils.degToRad(a)`

- converts angle *a* from degree to radians

`utils.identityMatrix()`

- returns an identity matrix, that performs no transformation

`utils.degToRad(a)`

- converts angle *a* from degree to radians

`utils.invertMatrix(M)`

- inverts matrix *M*

`utils.transposeMatrix(M)`

- transpoes matrix *M*

`utils.multiplyMatrices(M1, M2)`

- returns the product of matrices *M1* and *M2*

`utils.MakeTranslateMatrix(dx, dy, dz)`

- returns a translation matrix with displacement *dx, dy* and *dz*

`utils.MakeRotateXMatrix(a)`
`utils.MakeRotateYMatrix(a)`
`utils.MakeRotateZMatrix(a)`

- returns a rotation matrix of an angle *a* across the corresponding axis. The angle is specified in degrees.

utils.MakeScaleMatrix(s)

- returns a uniform scaling matrix of factor *s*.

utils.MakeScaleNuMatrix(sx, sy, sz)

- returns a non-uniform scaling matrix of factors *sx, sy and sz*.

utils.MakeShearXMatrix(hy, hz)
utils.MakeShearYMatrix(hx, hz)
utils.MakeShearZMatrix(hx, hy)

- returns a shear matrix with displacement *hx, hy or hz* across the corresponding axis.

You are invited to have a look at the implementation of the corresponding procedures contained in file **lib/utils.js** .