HW 04

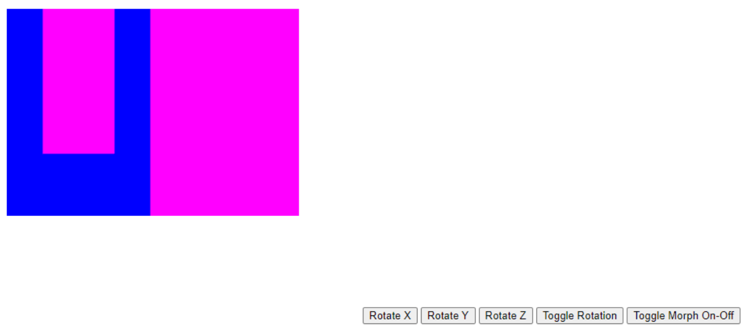
Write a (***I.***) WebGL application which will morph one 3D shape into another, and (***II.***) complete written problems based on the mathematics in transformations.

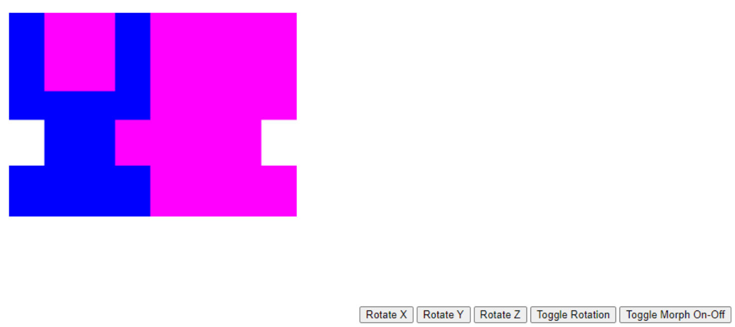
I. WebGL application

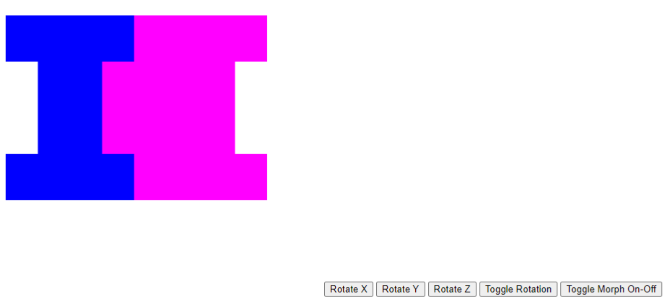
Morphing one point, ***P***, to another point, ***Q*** in 3D is based on the same concept as morphing in 2D:
1 – requires a parameter, ***t*** .
2 – ***t*** varies from 0.0 to 1.0. When ***t*** is 0.0, the returned value is ***P***, when ***t*** is 1.0, the returned value is ***Q***. If ***t*** is 0.5, the returned value is half of ***P***, and half of ***Q***.
3 – ***t*** * ***P*** + (1-***t***) * ***Q*** is equivalent to ***P*** + ***t*** * (***Q*** – ***P***)

It is probably best to define each object separately (but with the same number of vertices for the morphing), and create a buffer object for each, so they are both loaded into the vertex shader.

3D Morphing with Transformations, take your time, small steps, can define all polygons (no Transforms needed), refresh often while developing (2 objects, color consistent, then morph, then expand) written work. Written work, take your time multiplying matrices.







The morph parameter, ***t*** can be calculated in the application and sent to the vertex shader as a uniform variable.

In the vertex shader the morphing for the x and y positions can use the `.x` and `.y` references of the `in` variables.

The color can be morphed in the application and sent to the fragment shader as a uniform variable, but note the type will be ***4fv***, for each of R, G, B, and Alpha.

II. Written problems (the derivation in the Lecture Notes and in the textbook will be helpful for these)

1. Find the **three** tranformation matrices required for a rotation of 60 degrees (about the ***z-axis***) around the point (1,2,0,1), and then multiply these together to form one transformation matrix.

2. Show that a rotation (about the ***z-axis***) and a uniform scaling commute.
Hint: Setup the rotation matrix with a general angle of θ, $R_z(\theta)$ and the uniform scaling matrix with a general scale factor of α, $S(\alpha, \alpha, \alpha)$. Multiply these two matrices together, once with the rotation matrix first and once with the scaling matrix first. You should get the matrix for each of these products.

3. Show that two (general) translations commute.
Hint: find $T(x_1\ ,\ y_1\ ,\ z_1)$ and $T(x_2\ ,\ y_2\ , z_2)$. Multiply these two matrices together, once with one first, and then with the other first. You should get the matrix for each of these products.

4. Assuming we are interested in only two-dimensional graphics, we can use three-dimensional homogeneous coordinates by representing a point as $p = [x, y, 1]^T$ and a vector asv= $[a, b, 0]^T$. Find the 3 x 3 rotation, translation, and scaling matrices.