

### **HW 03**

Write a WebGL application which will morph one 2D shape into another.

To morph one point,  $P$ , to another point,  $Q$

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  $Q$ .

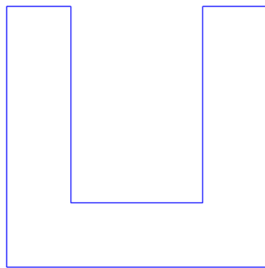
3 –  $t * P + (1-t) * Q$  is equivalent to  $P + t * (Q - P)$

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.

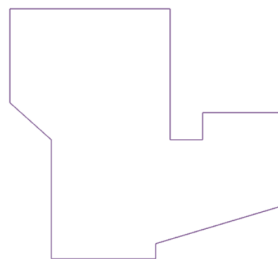
The morph parameter,  $t$  can be calculated in the application and sent to the vertex shader as a uniform variable.

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

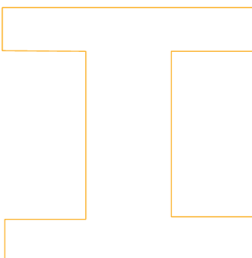
The color can be morphed in the application and sent to the fragment shader as a uniform variable, but note the type will be **4fv**, for each of R, G, B, and Alpha.



Toggle Morph On-Off



Toggle Morph On-Off



Toggle Morph On-Off