The scientific scenario: a quadrilateral representing continuous data needs to be displayed. Unfortunately, it is non-planar and the data values at the corner vertices map to four widely-varying colors.

How can we correctly smooth both the internal positions and colors, regardless of how we cut the quad into triangles?

Introducing the SuperQuad Geometry Shader!
Solution: Use bilinear interpolation to break the super-quad into sub-triangles.

\[ Q(s,t) = (1-s)(1-t)Q_0 + s(1-t)Q_1 + (1-s)tQ_2 + stQ_3; \]

\[ 0 \leq s, t \leq 1. \]

For any quantity, \( Q \), defined at the 4 vertices, \( Q \) can be interpolated into the interior with:

\[ Q(s,t) = (1-s)(1-t)Q_0 + s(1-t)Q_1 + (1-s)tQ_2 + stQ_3; \]