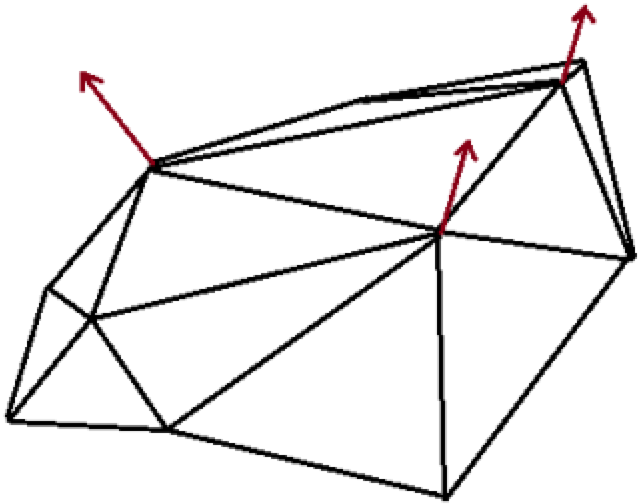
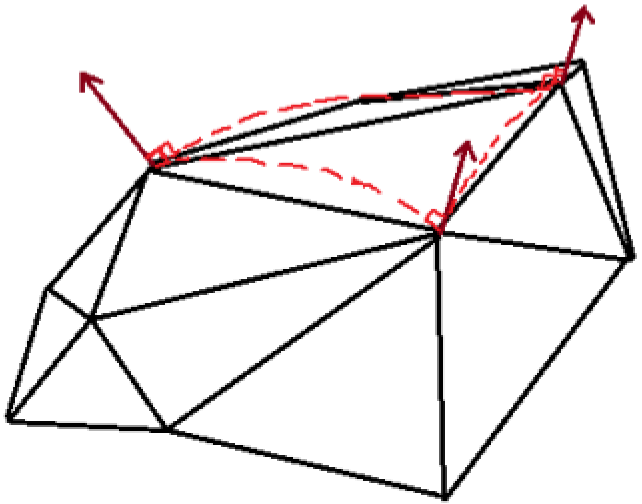
Barycentric Curve Interpolation From Simple 3D Geometry

and also cell shading

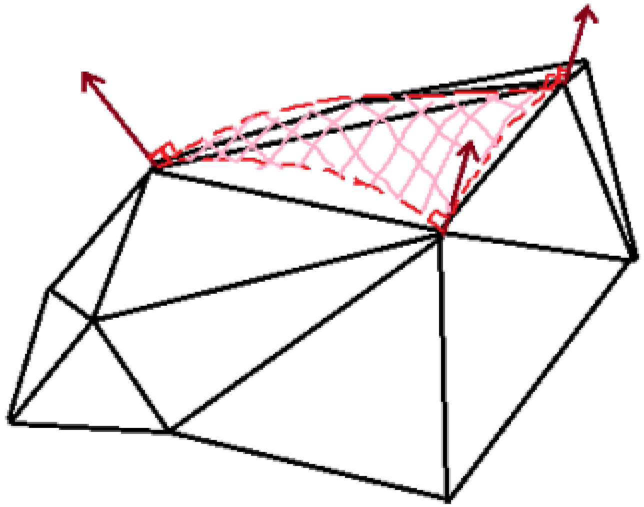
Project Overview



1 – Use vertex normal
like in Phong shading



2 – Create curves with the
perpendiculars to the
vertex normal along each edge

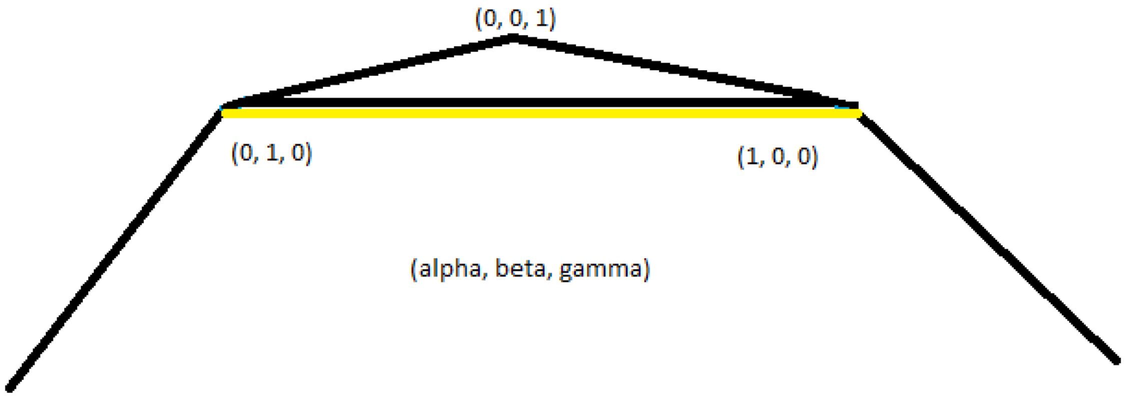


3 – Add ‘distance from face’ as
a Barycentric-derived parameter
and render

Barycentric in World Space

Consider the highlighted line:

- Gamma = 0 for the whole line
- Alpha = 1 – Gamma – Beta
- Can be determined as a function of just one variable



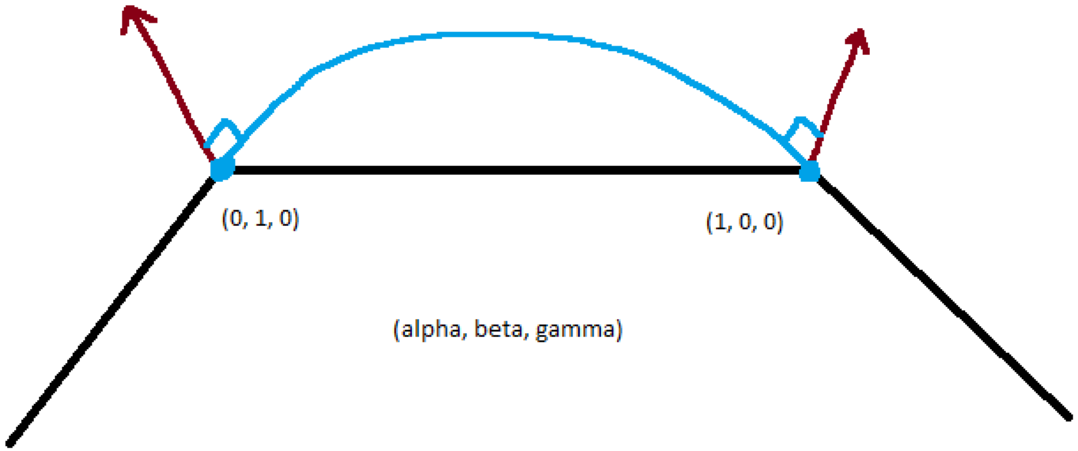
Equation of a line with 2 points and 2 slopes

$f(0) = 0$

$f(1) = 0$

$f'(0) = \theta$

$f'(1) = \Phi$



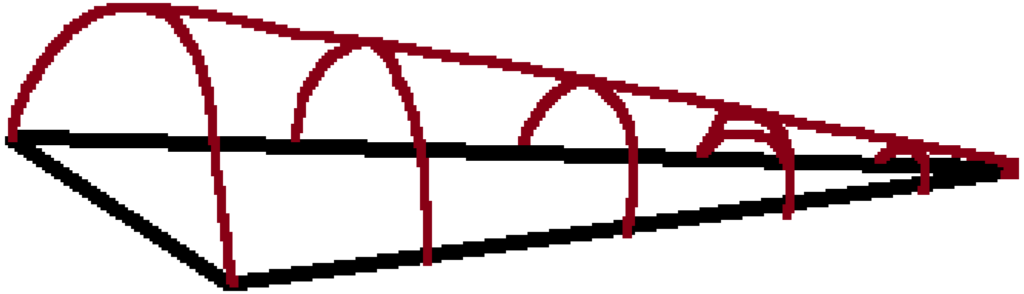
Hmmmmmm .... seems familiar to diffEq or beam bending

$$y = (\theta+\Phi)x\hat{}3 + (-2\theta-\Phi)x\hat{}2 + \theta x$$

But triangles aren't just their edges...

No longer mapping from alpha = 0 to alpha = 1

- Multiply alpha by 1/(1-opposing)



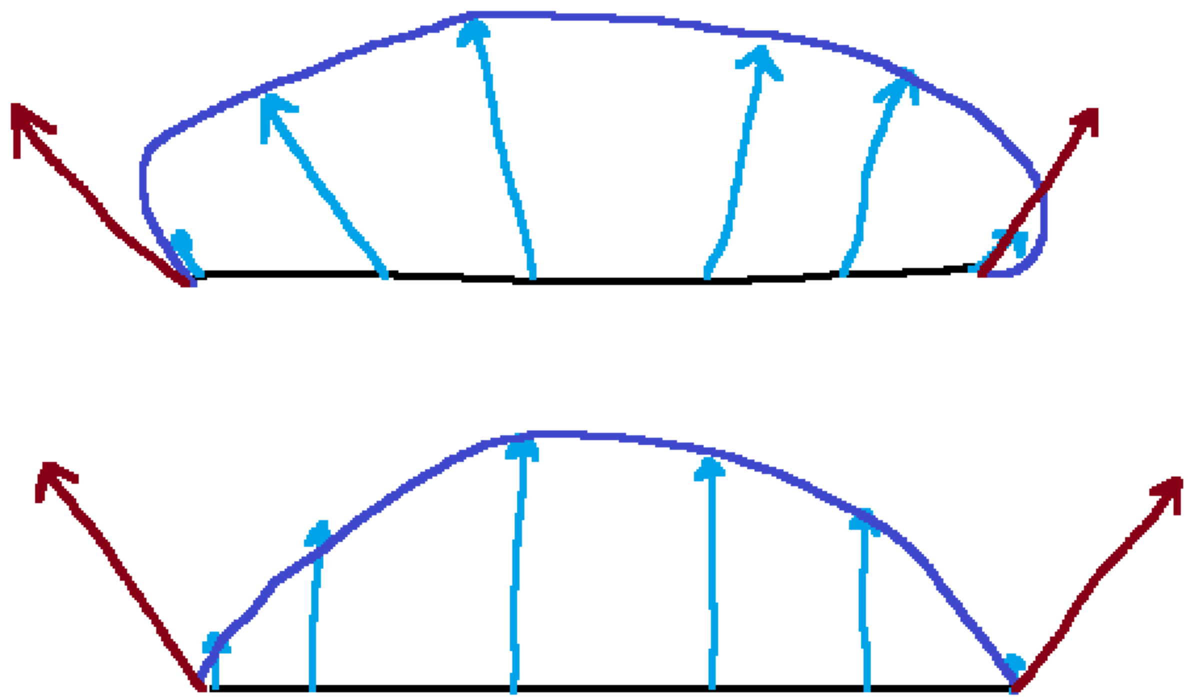
Function should be less important as it gets away from edge

- Multiply y by (1 – opposing)

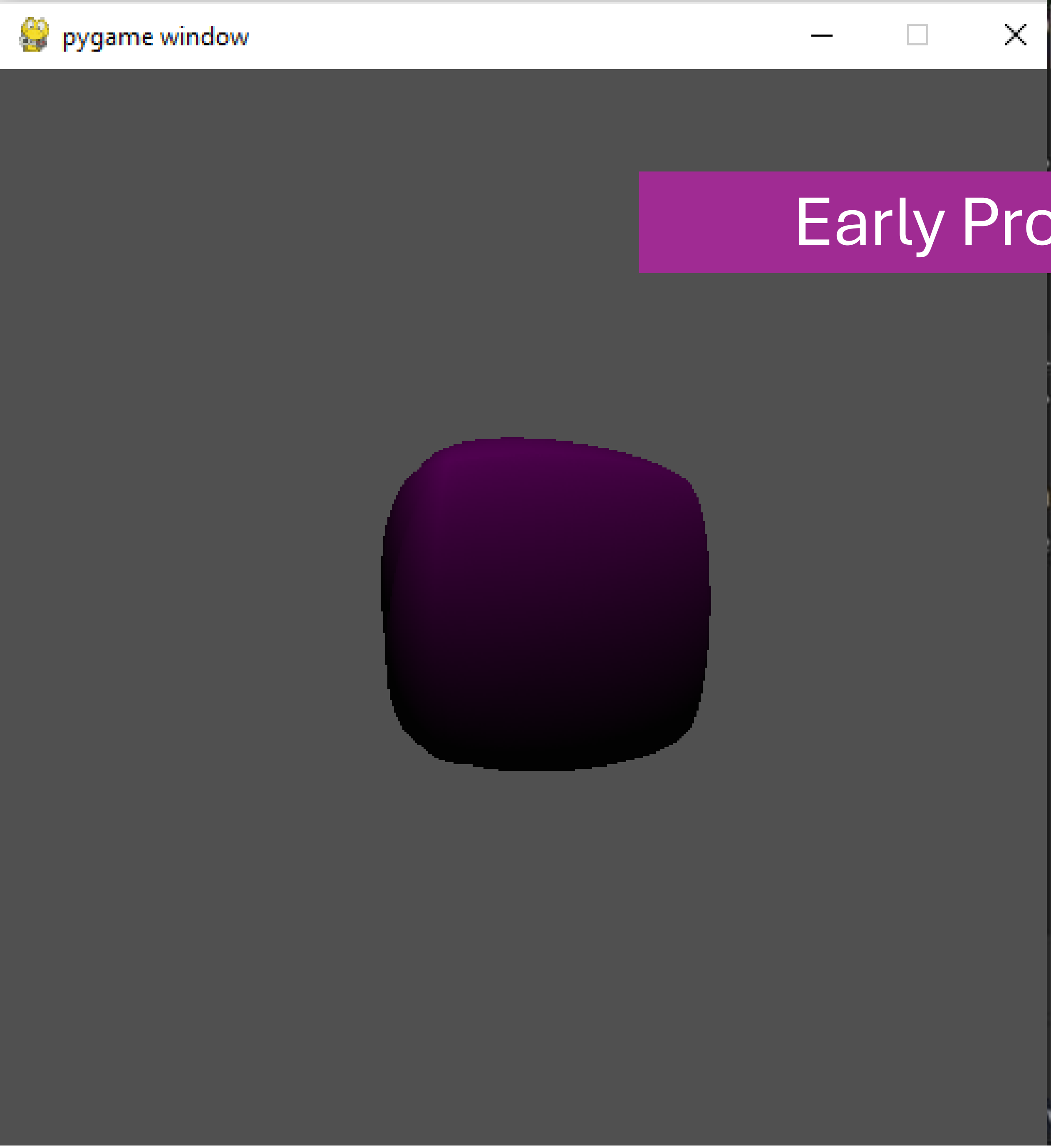
How to use 'y'

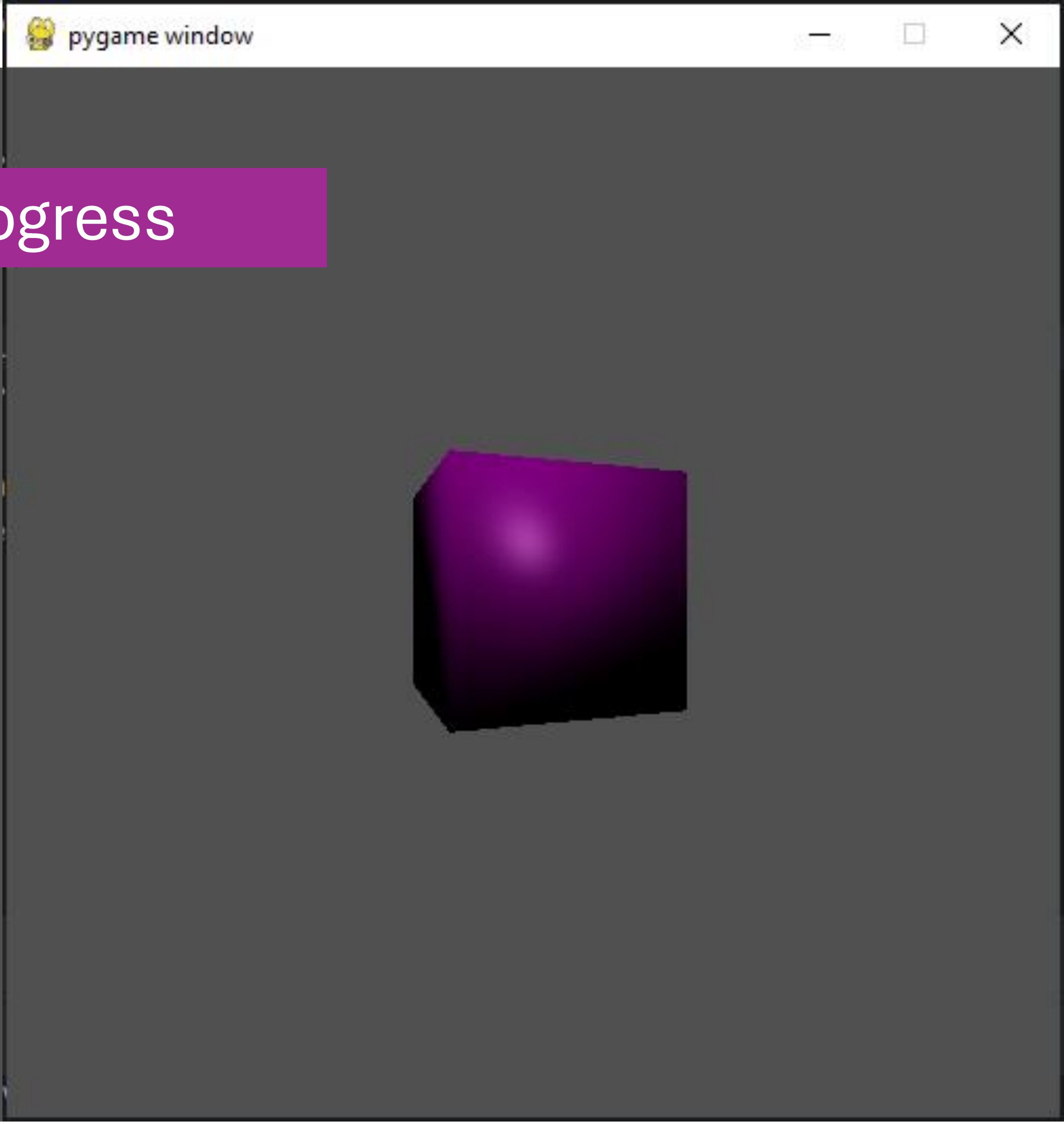
Applying to normal of point created a bubble

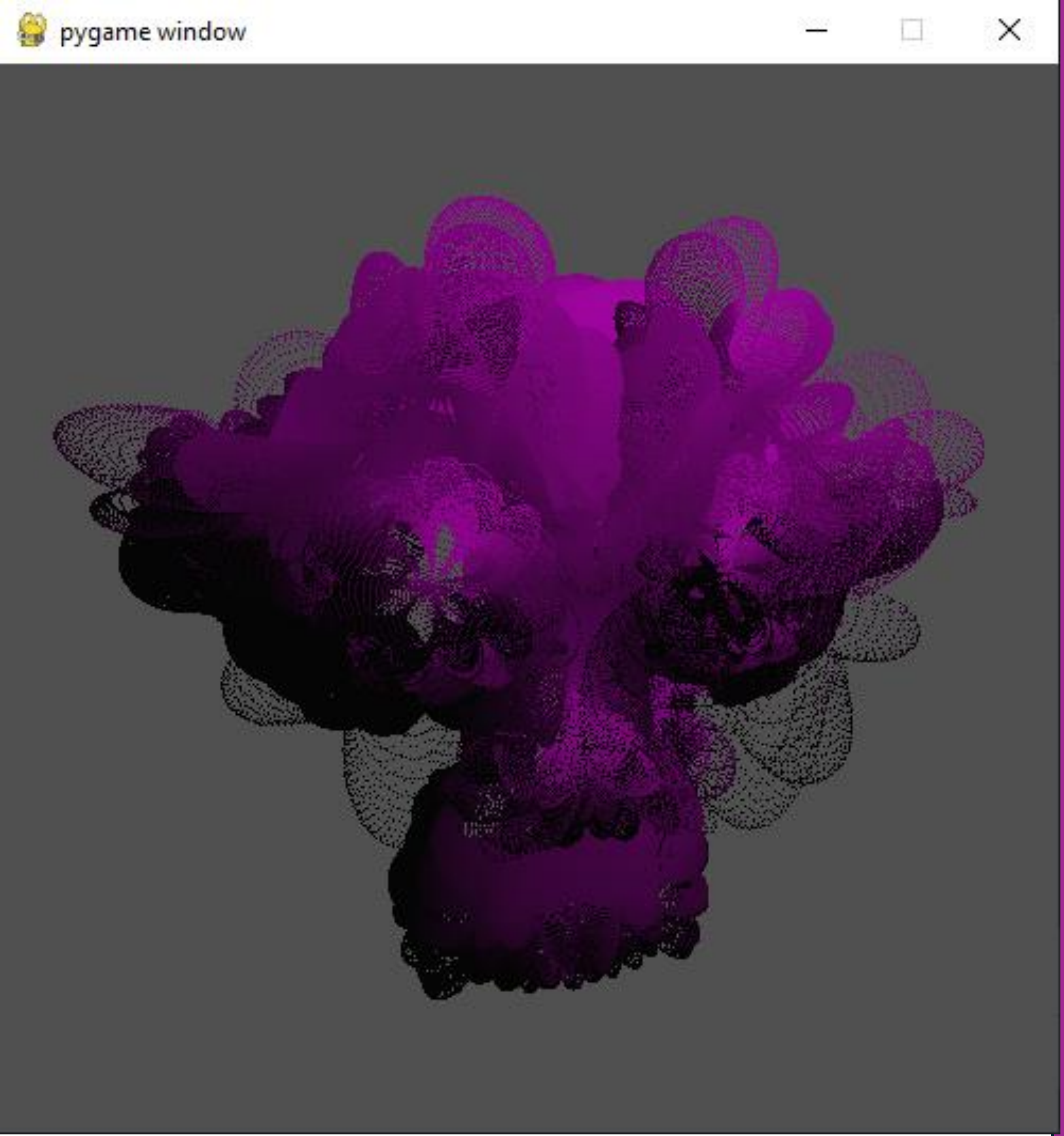
- apply it to the projection onto the perpendicular of the original line



Early Progress

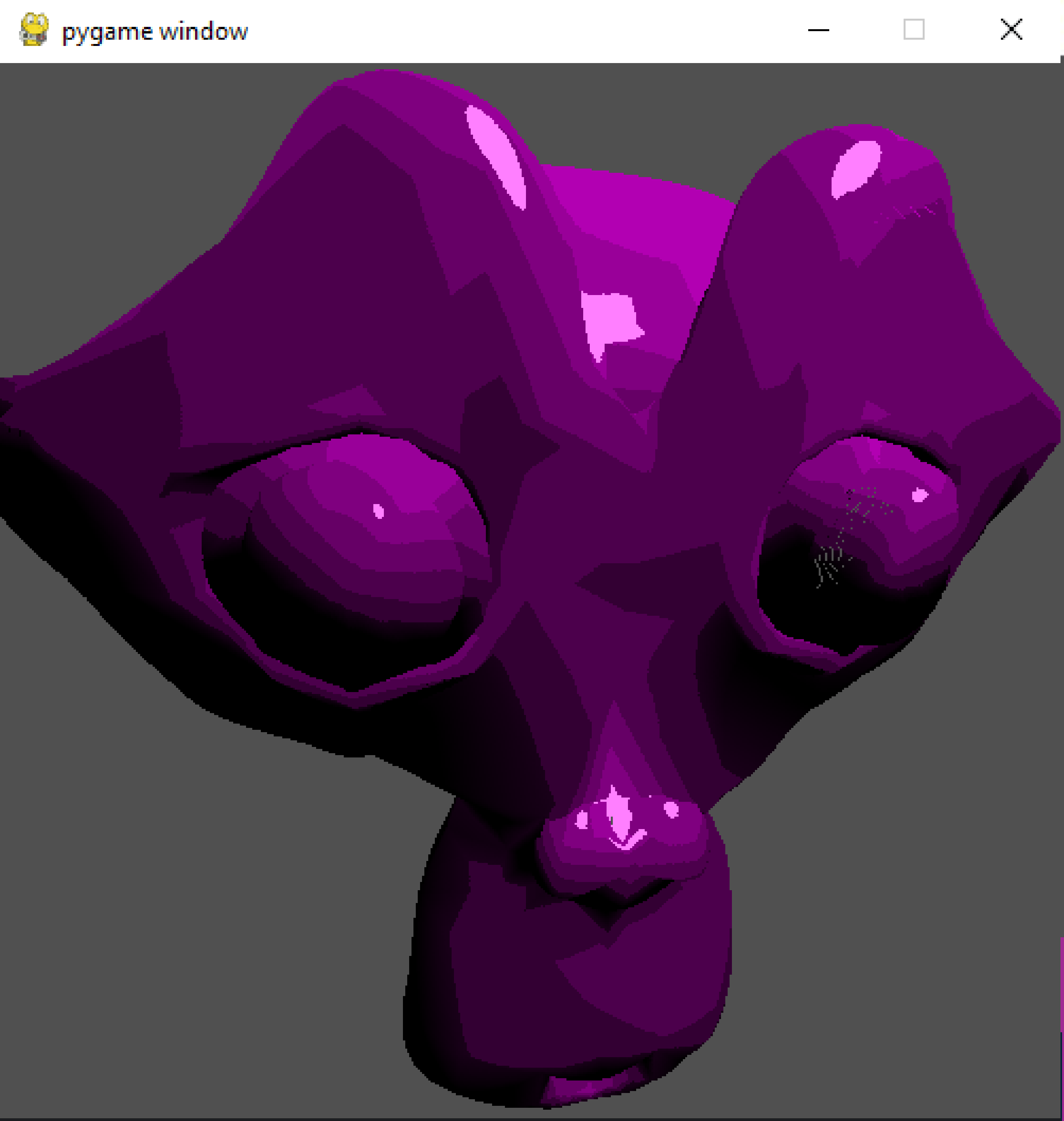


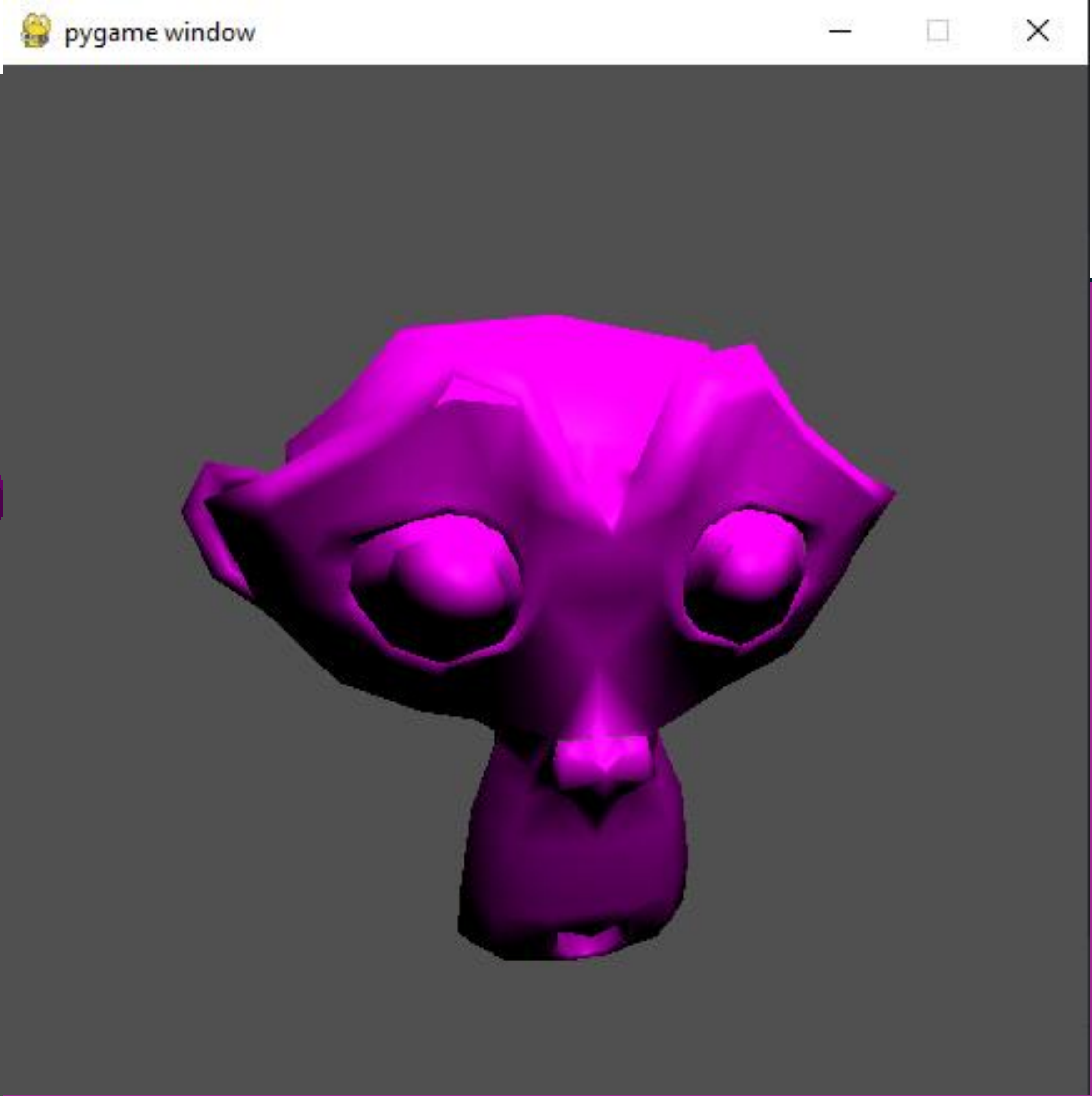




World-centered barycentric is worse for drawing than screen space barycentric

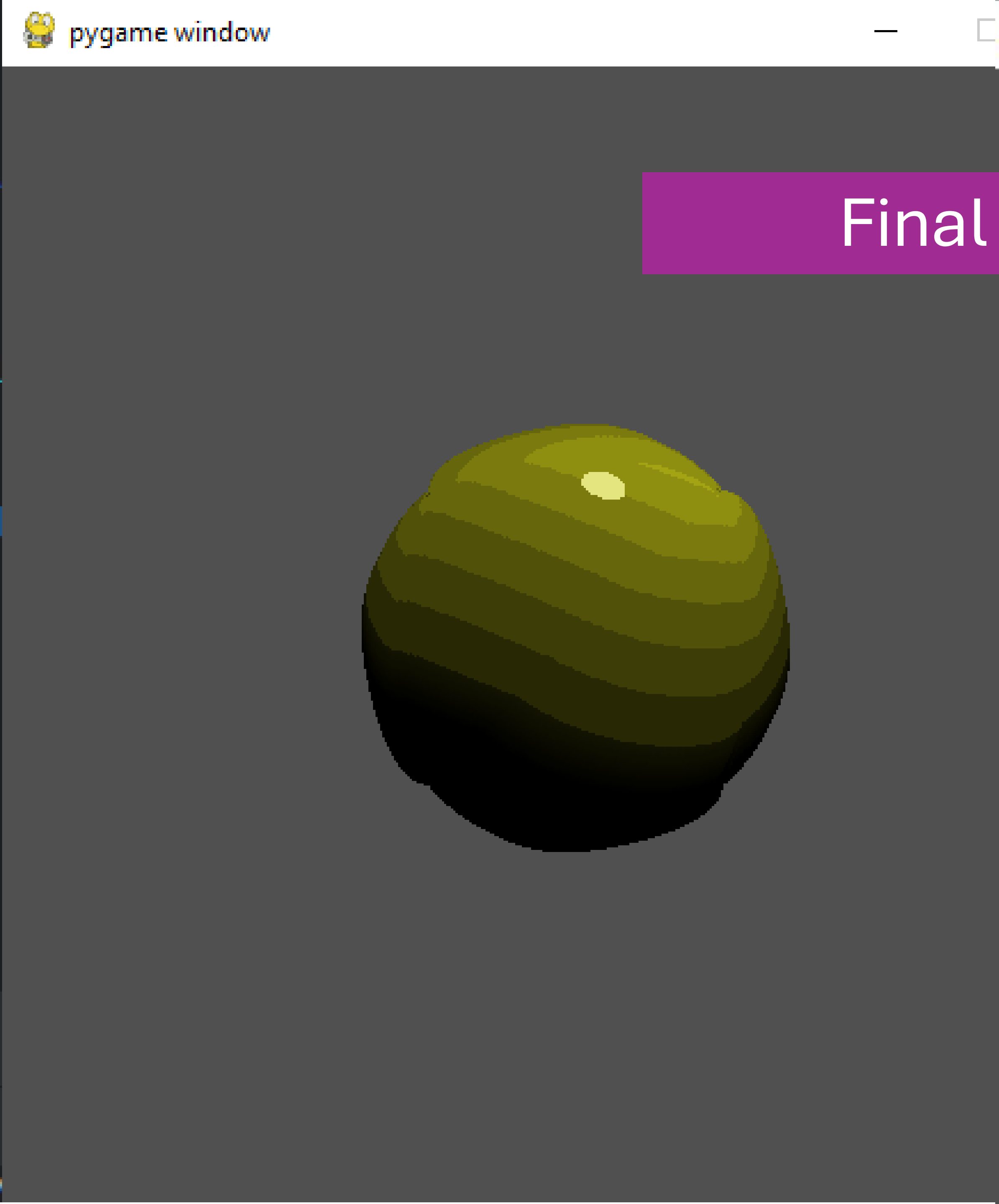
It can create perfect curves from simple geometry!

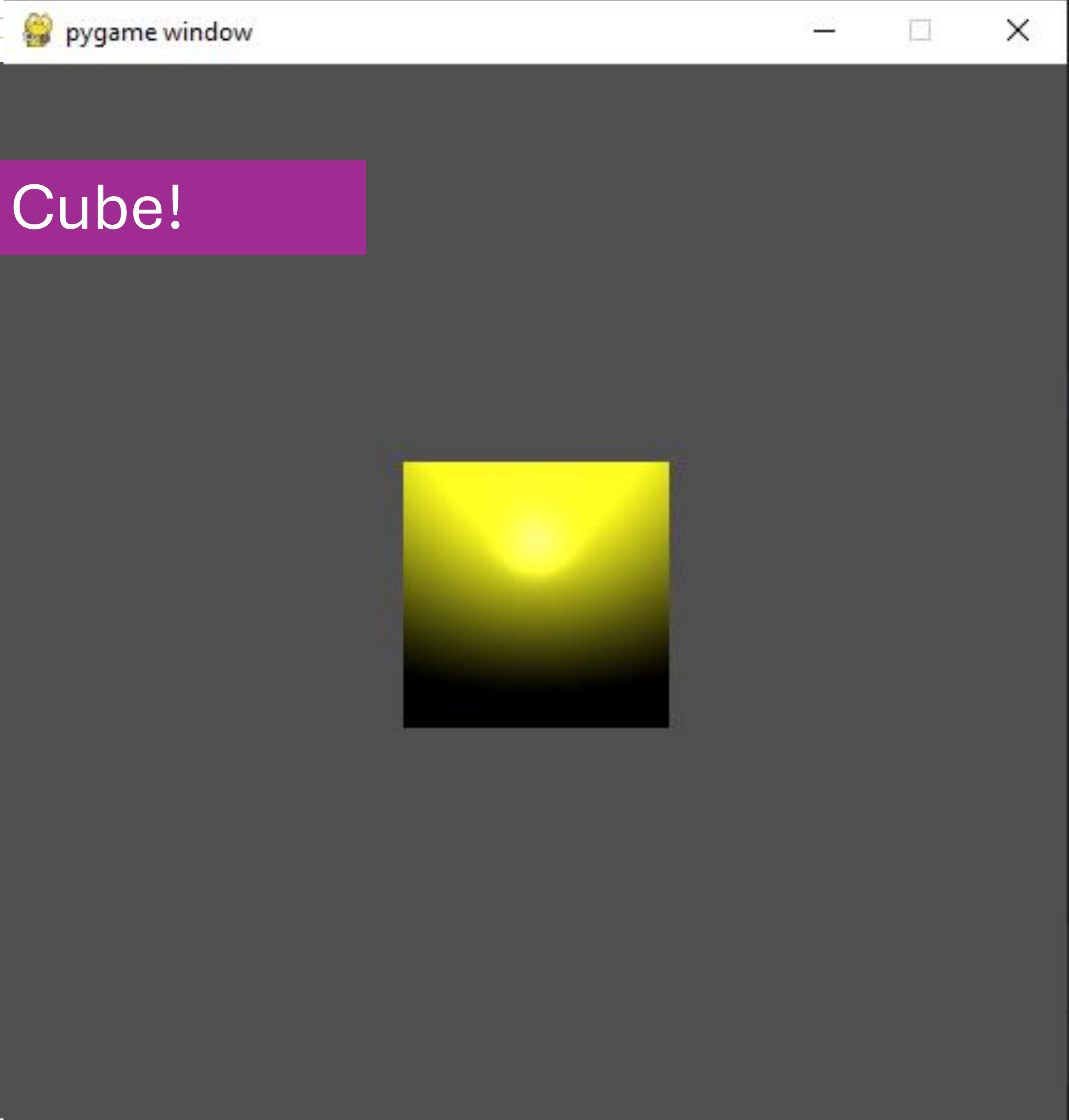




Suzanne in Phong versus mine

Final Cube!





Final Group Render!