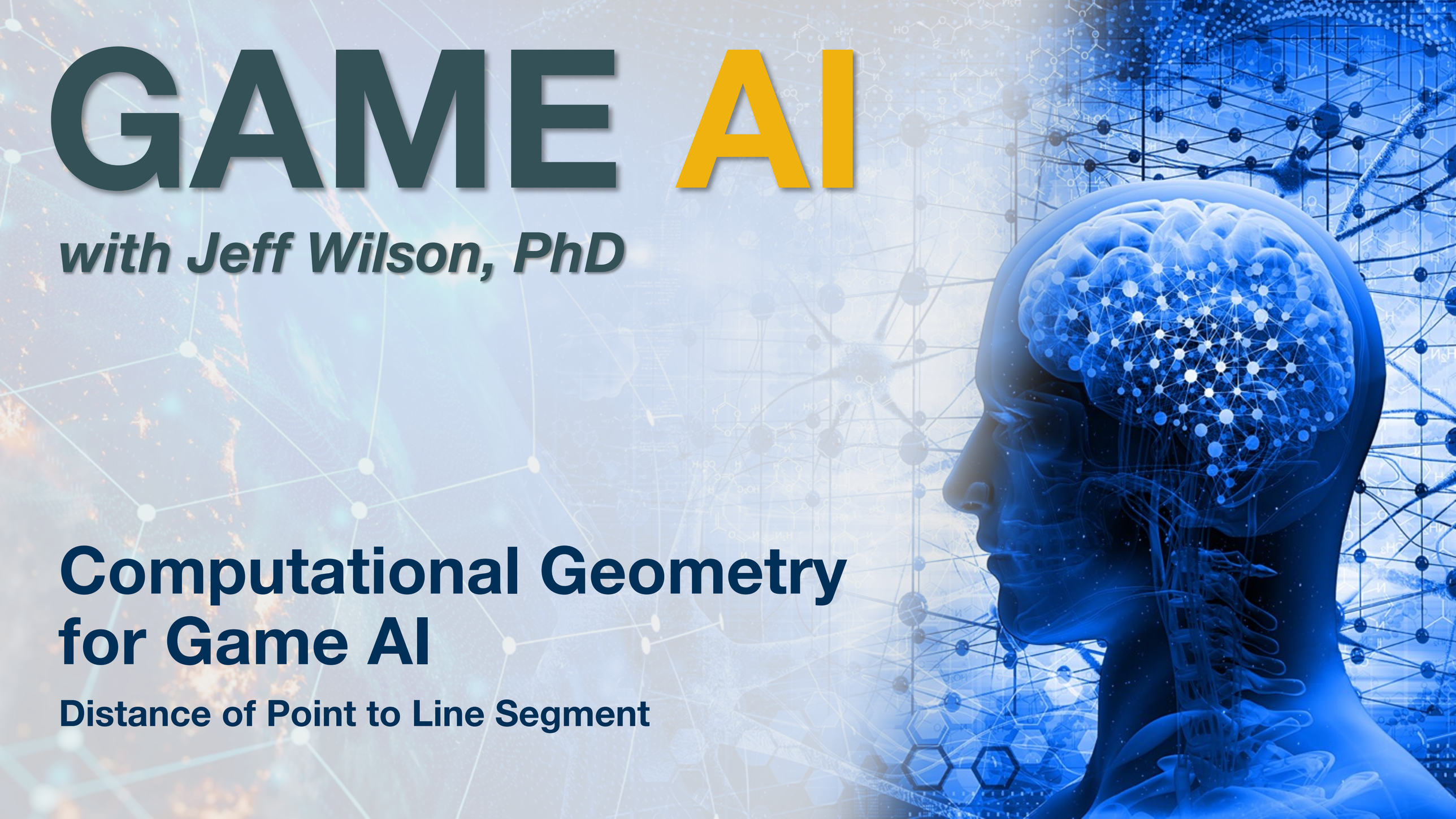
GAME AI

with Jeff Wilson, PhD

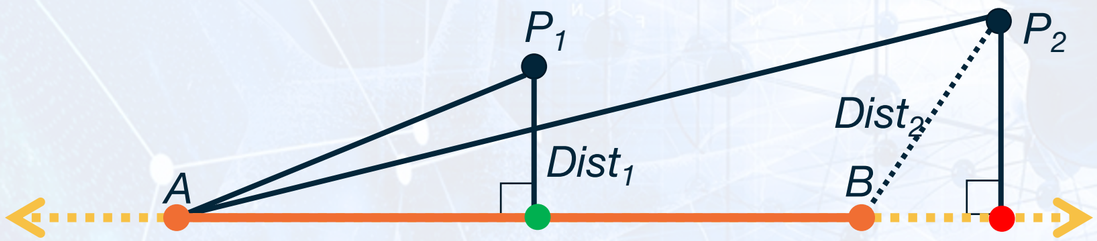


Computational Geometry for Game AI

Distance of Point to Line Segment

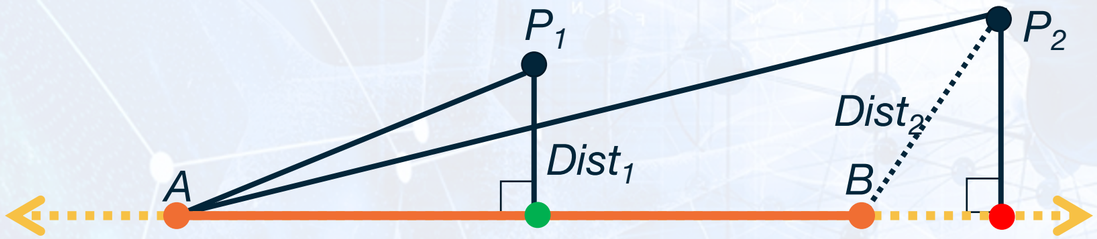
Distance of Point to Line Segment

- Floating point for calculation
  - Can cast mapped integers to floats or use original floats
  - Computational Geometry FP concerns are diminished but keep in mind for extreme cases with large exponents which could be encountered (consider PCG)



Distance of Point to Line Segment

- Algorithm has multiple steps
  1. Find closest point (*cp*) to *P* on the infinite line that is coincident with the line segment *AB* (via projection with *AP* or *BP*)
  2. Test whether *cp* is between *A* and *B* via vector line equation, checking scalar *t* in range [0, 1]. Is between?
     - YES – return distance between *P* and *cp*
     - NO – select from *A* and *B* which is closer to *cp*, then return distance from *P* to closer of the two

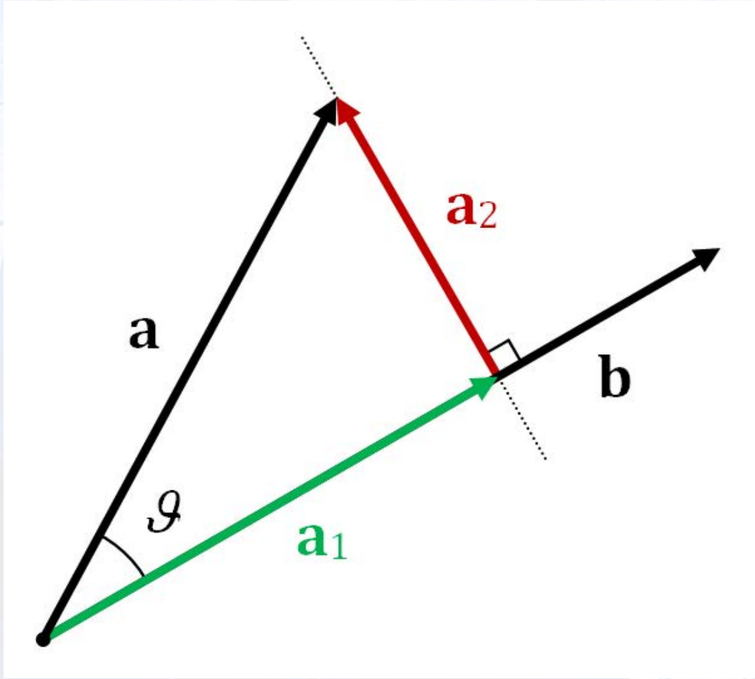


Distance of Point to Line Segment

- Nice optimization to avoid Square Root

$$\mathbf{a}_1 = a_1 \hat{\mathbf{b}}$$

$$a_1 = \|\mathbf{a}\| \cos\theta = \mathbf{a} \cdot \hat{\mathbf{b}} = \mathbf{a} \cdot \frac{\mathbf{b}}{\|\mathbf{b}\|}$$



$$\boldsymbol{a}_1 = \left(\boldsymbol{a} \cdot \frac{\boldsymbol{b}}{\|\boldsymbol{b}\|}\right) \frac{\boldsymbol{b}}{\|\boldsymbol{b}\|}$$

$$\boldsymbol{a}_1 = \frac{(\boldsymbol{a} \cdot \boldsymbol{b})\boldsymbol{b}}{\|\boldsymbol{b}\|^2}$$ ← Got rid of the Square Root!

And we have a normalized scalar that will be in range [0, 1] if on the line!

$$t = \frac{(\boldsymbol{a} \cdot \boldsymbol{b})}{\|\boldsymbol{b}\|^2} \qquad \boldsymbol{a}_1 = t\,\boldsymbol{b}$$

Error Checking

- Need to check if line segment points *A* and *B* are the same
- If so, return the distance from test point to either *A* or *B*
- Must do this to avoid divide by zero