Report for Project 2: Edge/Triangle intersection

Description of intersection calculation:

The intersection calculation was probably the hardest calculation to come up with. It involves finding a float t by means of various dot products involving the 2 lines which intersect. In essence, it is a parametric computation. The equation is something like the following: P(t) = A + tAB. A is the “starting point” for where the vector AB begins, and t is basically the % along AB that the intersection point will reside on. You also must take into account the other vector, which I call DE. Obtain the normal of DE. Then you can take the division of the dot product of the normal with DA, and the dot product of the normal with AB. The reason for the division is so that you can separate the parameter “t” from the rest of the equation, and find out what it is. By moving it to the other side of the equation, everything else turns negative. But now that you have “t”, you can plug it back into the parametric equation and use it for find the intersection point in real time.

Description of tolerance to force singularities:

The tolerance algorithm involves having a basic understanding of the geometric nature of the projection of a dot product between 2 vectors. It simple boils down to this: the dot product of a vector and it's normal will be 0, because the cos(90) is 0. Using this knowledge (and having help from the powerPoint slides) I constructed PinE, which draws a point if it is on the edge. But I allow the point to be drawn if the dot product of the line with its normal is within +-2 of 0, thereby expanding an area of tolerance by which to draw the (cyan) point when D or E are near to it.