Function `projPtToPlane[plane, p]`: For any arbitrary 3D point locates the nearest point on an arbitrary plane in 3 dimensions.
G. Palmer, Feb 20, 23, 24, 2015

Clear["Global `*""]

`projPtToPlane[plane_, p_] := Module[{x, y, z, p1, p2, p3, dotPrjToYCorner, normal, unitNormal, crossParallels, projPt, s},
   (* plane = {pt1, pt2, pt3}, where points of intersection with x, y, and z axes drawn with any origin. *)
   (* plane is arbitrary in 3D *)
   x = plane[[1]]; (* x, y, and z are points on the respective axes, e.g. {x1, x2, x3} *)
   y = plane[[2]];  
   z = plane[[3]]; 
   projPt = {p1, p2, p3};
   normal = Cross[z - x, z - y]; (* cross of two sides of triangle drawn between three axes *)
   unitNormal = normal/Norm@normal; (* unit vector from cross of sides *)
   (* cross of point-to-plane vector (p-projPt) with unit normal equals (0,0,0) giving three equations in p1, p2, p3 *)
   crossParallels = Cross[unitNormal, p - projPt] ;
   (* product line from projection point to corner (in plane) dotted to unit normal vector *)
   dotPrjToYCorner = (Dot[unitNormal, projPt - y] == 0); 
   (* We use only two of the equations from the cross of the unit vector with the point vector, because we need the corner dot product anyway. *)
   s = Solve[(crossParallels[[1]] == 0) &&
       (crossParallels[[2]] == 0) && dotPrjToYCorner, {p1, p2, p3}];
   {Flatten[projPt /. s], unitNormal} ]; (* module *)

pTxt[str_, pt_] := Text[Style[str, Bold, 8], pt];

Manipulate[
projPt = projPtToPlane[polyPts, p];

labels = {
    TXT["projPt", projPt[[1]] + {-0.2, 0.02, 0.02}],
    TXT["p", p + {0.03, 0.03, 0.03}],
    TXT["unit normal", {polyPts[[3]] + projPt[[2]]/2}]
};

Graphics3D[{FaceForm[LightBlue], Polygon[AppendTo[polyPts, polyPts[[1]]]],
    {Thick, Red, Arrow[{{p, projPt[[1]]}}]},
    {Thick, Black, Line[{{projPt[[1]], polyPts[[2]]}}]},
    {Thick, Black, Line[{{polyPts[[3]], polyPts[[3]] + projPt[[2]]}}]}, labels},
    Background -> White, Axes -> True, AxesLabel -> {"X", "Y", "Z"},
    PlotRange -> {{0, 1}, {0, 1}, {0, 1}}, ImageSize -> Medium, ViewPoint -> {2, -2, -3},
    Button["New projection", polyPts = {{Random[Real, {0.4, 1}], 0, 0},
        {0, Random[Real, {0.4, 1}], 0}, {0, 0, Random[Real, {0.4, 1}]}},
        p = {Random[Real, {0.5, 0.9}], Random[Real, {0.5, 0.9}],
            Random[Real, {0.5, 0.7}]}],
    TrackedSymbols -> {projPt, p}, SaveDefinitions -> True}]