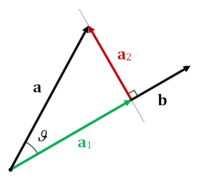
**Computer Graphics Notes M Feb 27**

See <http://www.cise.ufl.edu/class/cap4730sp15/pdf/12Object.pdf>

From <http://www.cise.ufl.edu/class/cap4730sp15/pdf/15Coordinates.pdf>

* Coordinates
  + Many different coordinate systems
  + Matrices are used to transfer btween them
  + Some coordinate systems
    - Texture Coords
    - World Coords
    - Camera (i.e. Projection)
    - Clip (i.e. Rasterization)
* Euclidean Space
  + Use the dot product (a.k.a. the scalar product) to determine
    - angles between vectors 🡺 cos(α) = (**v · w**) / (|**v**||**w**|)
    - length of a vector **v · v = v2 = vtv**
  + Use cross product to determine
    - Area **v** x **w**
    - Volume (**v** x **u**) **· w** (this is also the determinant)
  + Projections
    - Let be the projection of **v** onto **w**
    - Reflection of **v** across **w**: **v** − **2**
  + Transformations
    - Let M be the linear mapping of pi to pi’
    - M := [v1, v2, v1 × v2] ∈ **R**3×3
    - **vi := pi – p0, vi′ := pi′ − pi′**
    - **v3′** = M′M-1 **v3** ∈ **R**3
    - Exercise: Parameterize a rotation about a line in 3-space.
      * When rotating about arbitrary axis, transform the axis such that it lines up with either the x-, y-, or z-axis (major axes). First translate the arbitrary axis and the object, such that the arbitrary axis goes through the origin, then rotate the arbitrary axis and the object such that the arbitrary axis is a lined with a major axis, then rotate the object about the arbitrary axis, and undo the rotation and translation done to the arbitrary axis and the object.
      * T is the translation matrix
      * Rx and Ry are rotation matrices about the x- and y-axis respectively
      * **b** is a point on our object
      * T-1 Rx TRyT(arbitrary rotation matrix) RyRxT **b**
      * **Read up on: quadrics, homogenous cords & projective space, Bezout’s Thm**