# Vector Math Review

Spend about an hour reviewing the following topics, focus on application of the concepts in Computer Graphics

Follow any standard text will do, or, an excellent interactive resource for students

<http://chortle.ccsu.edu/VectorLessons/vectorIndex.html>

# Points, and Vectors

## Vector: Column vs Row notation

## Addition, Subtraction

## Size

## Scaling vectors

## Normalization and Direction (unit vectors)

# Vectors and Lines

## Parametric Line

* Given points P1, P2
* d = P2 – P1
* D = normalized(d)
* Line: P1 + tD (D normalized!)

## Distance along the line

* t < 0: behind
* t > 0 and t< Distance(P1,P2)
* t > Distance(P1, P2)

# Dot Product

## Definition and properties

* 0 dot U = 0
* U dot V = U dot V
* U dot U = length(u)-squared
* Dot of normalized vectors is cosine
  + Dot = 0 => Perpendicular
  + Dot > 0 => angle less than 90 (“similar direction: front/back”)
  + Dot < 0 => angle more than 90 (“opposite direction”
  + A word of warning against testing for floating point value
  + E.g., (dot(U,V) == 0.0f)!!
* aU dot V = a (U dot V)
* (aU + bV) dot W = (aU dot W) + (bV dot W)

## Projected Length

* Emphasize projected direction must be a unit vector

## Applications:

* Distance between a point and the line
  + Use in selection of lines/objects
* Reflection of a line
  + Use in computing mirror reflection
  + Simple plane equation representation

# Cross Product

## Definition and properties

* Good tutorial: <http://www.phy.syr.edu/courses/java-suite/crosspro.html>
* W = U x V
  + W is perpendicular to both U and V
  + True in any dimension!
* U x V = -V x U
* U x U = 0
* (U x V) x W != U x (V x W)
* U x (V + W) = U x V + U x W

## Orthogonal vectors

## Cartesian as a special case of orthonormal base

* In 2D: (x,y) as Vector: xi + yj

## Computing orthonormal set

* A rectangle is a 2D space in 3D world
* Three non-colinear points span 2D space: Tripod
* Three points: 2 independent lines
* Cross to find normal, cross with normal to find side vector

## Applications

* Orthonormal base at the camera