2D Vector Math for Games

http://cse.unr.edu/~moberberger/2dvector.pdf

C# Source Code:

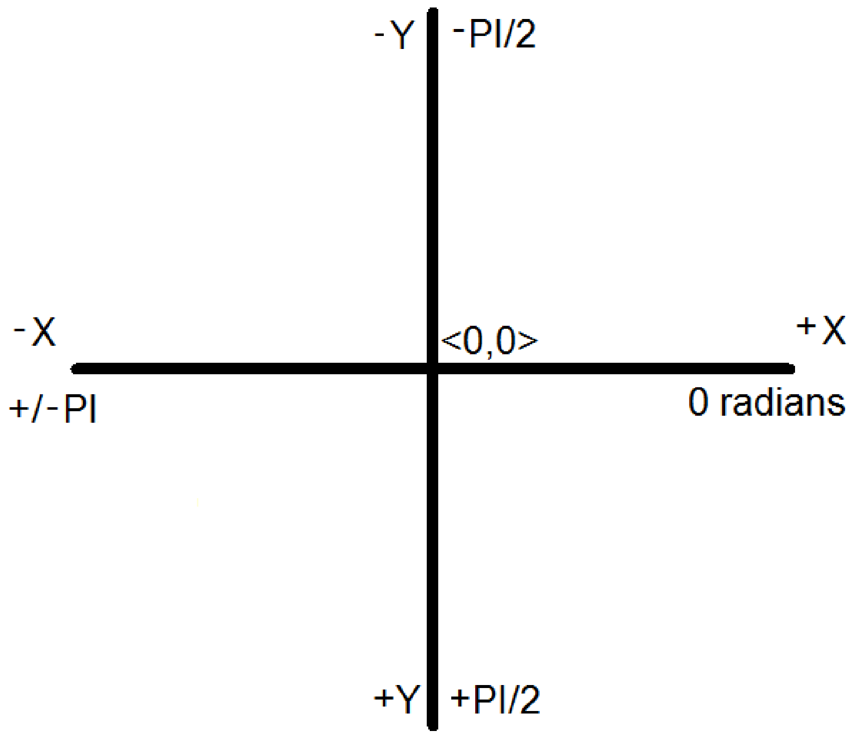
http://cse.unr.edu/~moberberger/2dvector.zip

Transformation

- We are talking about planar spaces
- 3d spaces need to be transformed
  - For instance, <x,y> = <-a,0,b>
  - There are infinite numbers of possible transformations
    - May include rotations
    - May include scaling
    - Probably won’t require translations

Planar Coordinates

- We will use the following planar coordinate system:



Basic 2d Vector Operations

- Vector Addition (and implicitly subtraction)
- Scalar Multiplication (division, negation)
- Magnitude (vector length)
- Unit Vectors (magnitude, division)
- Vector Comparison (FP precision errors)
- Angle Conversion (to/from radians)
- Dot Product

Variables

- Uppercase: Vector Lowercase: Scalar
- <x,y> - A Vector comprised of Scalar x and y
- Vectors- P: Point, V: Velocity
- Scalars- h: Heading, s: Speed

- $D = P_2 - P_1$
  - D is a vector from P_1 to P_2
  - $|D|$=Distance between P_1 and P_2

Angle Conversion

- Basic Trigonometry – RADIANS!
- From Angle to Vector:

  x = cos(h)   y = sin(h)

  <x,y> is a unit vector, say V_U: V = V_U *s for Velocity
- From Vector to Angle

  h = atan2(y, x)

  s = length(<x,y>)

$$\operatorname{atan2}(y,x) = \begin{cases} \arctan(\frac{y}{x}) & x > 0 \\ \pi + \arctan(\frac{y}{x}) & y \ge 0, x < 0 \\ -\pi + \arctan(\frac{y}{x}) & y < 0, x < 0 \\ \frac{\pi}{2} & y > 0, x = 0 \\ -\frac{\pi}{2} & y < 0, x = 0 \\ \text{undefined} & y = 0, x = 0 \end{cases}$$

Desired Heading and Speed

- We approximate acceleration
- You know: $t, P, V, \Delta s_{max}, \Delta h_{max}, s_{des}, h_{des}$ THEN:

$h = atan2(V_y, V_x)$, $s = |V|$

$\Delta h = h_{des} - h$

IF $abs(\Delta h) > \Delta h_{max}$: $\Delta h = sign(\Delta h) * \Delta h_{max}$

$\Delta s = s_{des} - s$

IF $abs(\Delta s) > \Delta s_{max}$: $\Delta s = sign(\Delta s) * \Delta s_{max}$

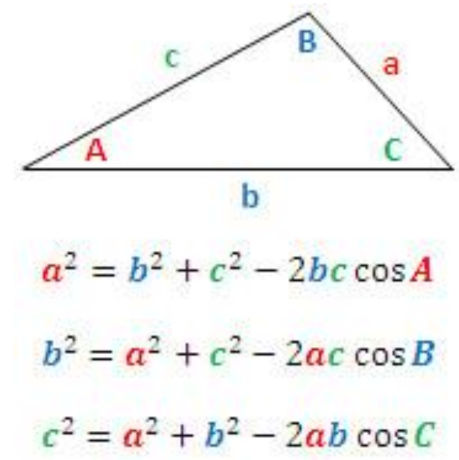
$s \mathrel{+}= t\Delta s$ $h \mathrel{+}= t\Delta h$ *make sure:* $-\pi <= h <= \pi$

$$V = s * \langle cos(h), sin(h) \rangle$$

Moving towards a Point

- Vector to the target: $D = P_{target} – P$
- Desired Angle to target: *atan2*(D_y, D_x)
- Desired Speed:
  - To reach target in 1 time unit, use |D|
  - Check against your “maximum speed”

Dot Product



- Analogous to the Law of Cosines

  $c^2 = a^2 + b^2 - 2ab\cos(\theta)$

- Dot Product is a scalar value

  $A \cdot B = A_X * B_X + A_Y * B_Y$

  $A \cdot B = |A||B|\cos(\theta)$

- Rearranged

  $\cos(\theta) = (A \cdot B) / (|A||B|)$

  $\theta = \cos^{-1}((A \cdot B) / (|A||B|))$

- *Very useful for Interception of Moving Objects*

Interception of Moving Objects

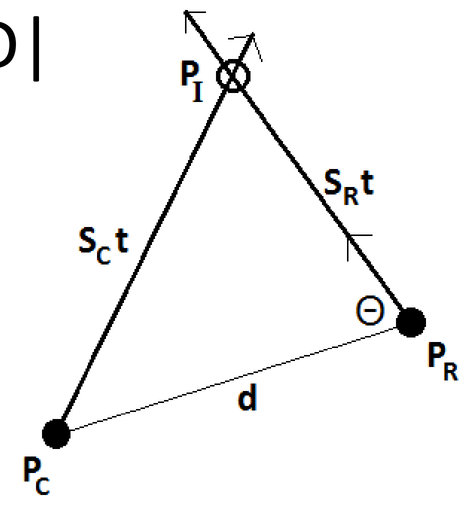
- Things We Know about Coyote and Roadrunner

  P_C, P_R, V_R, s_C: Positions, Tgt Velocity and My Speed

  t = time, $s_R = |V_R|$, $D = P_C – P_R$, $d=|D|$

  P_I = Point of Interception

  $cos\theta = (V\cdot D) / (ds_R)$



- Law of Cosines tells us:

  $(s_Ct)^2 = (s_Rt)^2 + d^2 - 2s_Rtdcos\theta$

This reduces to a Quadratic Equation in 't'

Interception Continued

- Using simple algebra, the equation becomes:

  $(s_C^2 - s_R^2)t^2 + (2ds_R \cos\theta)t - d^2 = 0$

  $a = (s_C^2 - s_R^2)$ $b = (2ds_R \cos\theta)$ $c = -d^2$

  You know all of these values already, even *cos*ϴ!

- Solving the quadratic, you will get t_1 and t_2

  Set ‘t’ to the smaller positive value of t_1 and t_2

  $P_I = P_R + V_R t$

  Set desired heading based on “Moving Towards a Point” using P_C and P_I

Interception Continued

- Degenerate Cases:
  - You are already at your target
  - Your target's speed is zero
    - If your target is not moving, then your target IS your intercept point
- Cases Preventing Interception
  - Your max speed is zero- you cannot move
  - The Quadratic cannot be solved
    - Or, it can be solved but both t_1 and t_2 are negative