Problem:
Given a point \( p \), a vector \( v \), and a circle defined by a center \( c \) and a radius \( r \). All in 2D.
Give a Boolean answer to the question:
Does the ray defined by \( p \) and \( v \) intersect the circle defined by \( c \) and \( r \)?
The ray is the half line defined by \( r(t) = p + tv \) with \( t \) in [0,\( \infty \)].

Solutions:
Part 1: Describe two solutions to the problem. Here’s three approaches.
A) Substitute the equation for the ray into the equation for a circle and solve for \( t \).
\[ r(t) = p + t v = (px + t vx, py + t vy) \]
gives us \( x \) and \( y \) in terms of \( t \).
Substitute these \( x \) and \( y \) values into the circle equation
\[ (x-xc)^2 + (y-yc)^2 = r^2 \]
If \( t \) has a positive real solution, hit.
B) Compute the angle \( \theta \) between the vectors \( v \) and \( u=(C-p) \), and project \( u \) onto the unit vector in the direction of \( v \) to get \( x \). Then you have the equation
\[ \tan(\theta) = y/x, \]
solve for \( y \) which is the distance from the line to \( C \), and if \( y < r \), it’s a hit.
C) Project the vector \( u=(C-p) \) onto a unit length normal to \( v \), and then take its magnitude. Compare this magnitude to \( r \), and if it’s less, hit.

Part 2: Give one solution in detail.
This is the solution to C, which is more or less the "official" solution as we work with vectors. This approach uses the perp vector – a similar solution starts with orthogonal projection (but normalizes \( v \) beforehand.)

Steps:
1) Given \( v=<x,y> \), compute the perpendicular vector \( v_{\text{perp}} = <-y,x> \)
2) Normalize \( v_{\text{perp}} \) to get \( v_{\text{perp normalized}} \).
3) Compute the vector \( PC = C - p \).
4) Take the dot product \( d = PC \cdot v_{\text{perp normalized}} \)
5) If \( d < r \), it’s a hit.
Notes on student solutions

Everyone had reasonable solutions, and there were some very solid, imaginative solutions, beyond the three above. The class as a whole did solid work. But, could do better. If you can give a really strong, verified and efficient solution to this problem, you have

Not many points were taken off this time for incorrect answers, but

1) Many had the vector v projected onto the vector PC, everything else right. That gives the bold vector below which is not the shortest distance from the center to the line defined by p and v. This error was done in many different approaches, not just orthogonal project. One lesson from this is that when we did orthogonal project in class, we projected the upper vector onto the lower. So many applied that pattern with thinking which vector should be which.

2) Others took the orthogonal projection as done in class, and didn’t normalize the v_perp vector (or u2 in the orthogonal procedure). This gives a distance that’s too long, but an understandable mistake since we didn’t discuss it in class. Blame the instructor. No points taken off.

3) In solution 1A, substituting the line equations into the circle equation and solving for an intersection solution, several students used the line equation y = mx + b (so the point would (x, mx+b).) That fails when the line is vertical (with m = deltaY/0). Almost always better to use the parametric equation (px+t*vx, py+t*vy).

Only one student in the entire class tested their method with actual data. Threw in a real line and circle. Had all of you done that you would not have submitted so many “almost” solutions. You do unit testing on projects – think of doing them on pseudocode, and don’t always wait for the instructor to give them. Octave-online lets you do the calculations quickly. A simple example – like point-vector p=(0,0), v=(1,1), circle = (2,0) with radius 1 (or radius 5) – would allow you to verify your equations.

4) Similarly, no one compared their solutions for efficiency. For game design and graphics in general, shaving milliseconds off with efficiency can be essential. Of the right solutions everyone submitted, some are very efficient and others, while correct, not the most efficient.