Moving circle vs static line segment problems and solutions

Problem:

We have the following line segment:

LNS: [P0(62, 10) , P1(-64, 7)]

We have the following circle:

CIRCLE: [B_s(-235, 88) , R = 15]

Moving in this game loop with a vector V(100,-50). (where V is the vector B_sB_e)

You need to test if CIRCLE will collide with LNS.

If there is collision, what would the position of the circle be at intersection time?

If any intersection, consider a bouncy circle, what would the final position of the circle be, after collision and reflection?

Solution:

//Compute the outward normal N of LNS:

P0P1(-64 – 62, 7 – 10) = P0P1(-126, -3)

N(P0P1.y, - P0P1.x) = N(-3, 126)

Normalized N = (-3 / 126.0357, 126 / 126.0357) = (-0.0238, 0.9997)

//Compute N.Bs – N.P0

Distance = N.Bs – N.P0 = (-0.0238, 0.9997).(-235, 88) - (-0.0238, 0.9997).(62, 10) = 5.593 + 87.9736 + 1.4756 - 9.997 = 85.0452

Distance is > (+R = 15)

//Check if the velocity vector V is within the end points of the imaginary line LNS2, that is parallel to LNS and distant by +R

//Compute M which is the outward normal of V

M(V.y, - V.x) = M(-50, -100)

//Compute end points of LNS2

P0' = P0 + RN = (62, 10) + 15*(-0.0238, 0.9997) = (62, 10) + (-0.357, 14.9955) = (61.643, 24.9955)

P1' = P1 + RN = (-64, 7) + 15*(-0.0238, 0.9997) = (-64, 7) + (-0.357, 14.9955) = (-64.357, 21.9955)

//Compute M.BsP0' * M.BsP1'

M.BsP0' * M.BsP1' =
(-50, -100).((61.643, 24.9955) - (-235, 88)) * (-50, -100).((-64.357, 21.9955) - (-235, 88)) =

(-50, -100).(296.643, -63.0045) * (-50, -100).(170.643, -66.0045) =

(-14832.15 + 6300.45) * (-8532.15 + 6600.45) = -8531.7 * -1931.7 = 16480684.89 > 0

//The result is > 0
//We will call "**CheckMovingCircleToLineEdge(false)**"

//Check which edge may collide first; is it P0 or P1?

//Compute normalized M

Normalized M = (-50 / 111.80, -100 / 111.80) = (-0.447, -0.894)

dist0 = BsP0.M = (62-(-235),10-88).(-0.447, -0.894) = (297,-78).(-0.447, -0.894) = -132.759 + 69.732 =
-63.027

dist1 = BsP1.M = (-64-(-235), 7-88).(-0.447, -0.894) = (171, -81).(-0.447, -0.894) = -76.437 + 72.414 =
-4.023

abs(-63.027) > R //Reject

abs(-4.023) < (R = 15) //possible collision - we choose P1 side

Normalized V => gotten from Normalized M = $\hat{V}$(0.894, -0.447)

float m = BsP1.V = (-64 - (-235), 7 - 88).(0.894, -0.447) = (171, -81).(0.894, -0.447) = 152.874 + 36.207 =
189.081 > 0 // V facing LNS => possible collision

//Compute s = sqrt(R*R – dist1*dist1)

s = sqrt(15*15 - (-4.023)*(-4.023) = 14.45

//while normalizing M, earlier, we computed the length which is = V.Length() = 111.80
float ti = (m – s) / V.Length() = (189.081 - 14.45) / 111.80 = 1.5619 > 1

//Conclusion

The Circle will not reach the line segment LNS in this frame. No reflection will happen.