CSD1130
Game Implementation Techniques

Lecture 17b

Overview

- Animated Circle to Line Segment

Modeling Pinball as Circle

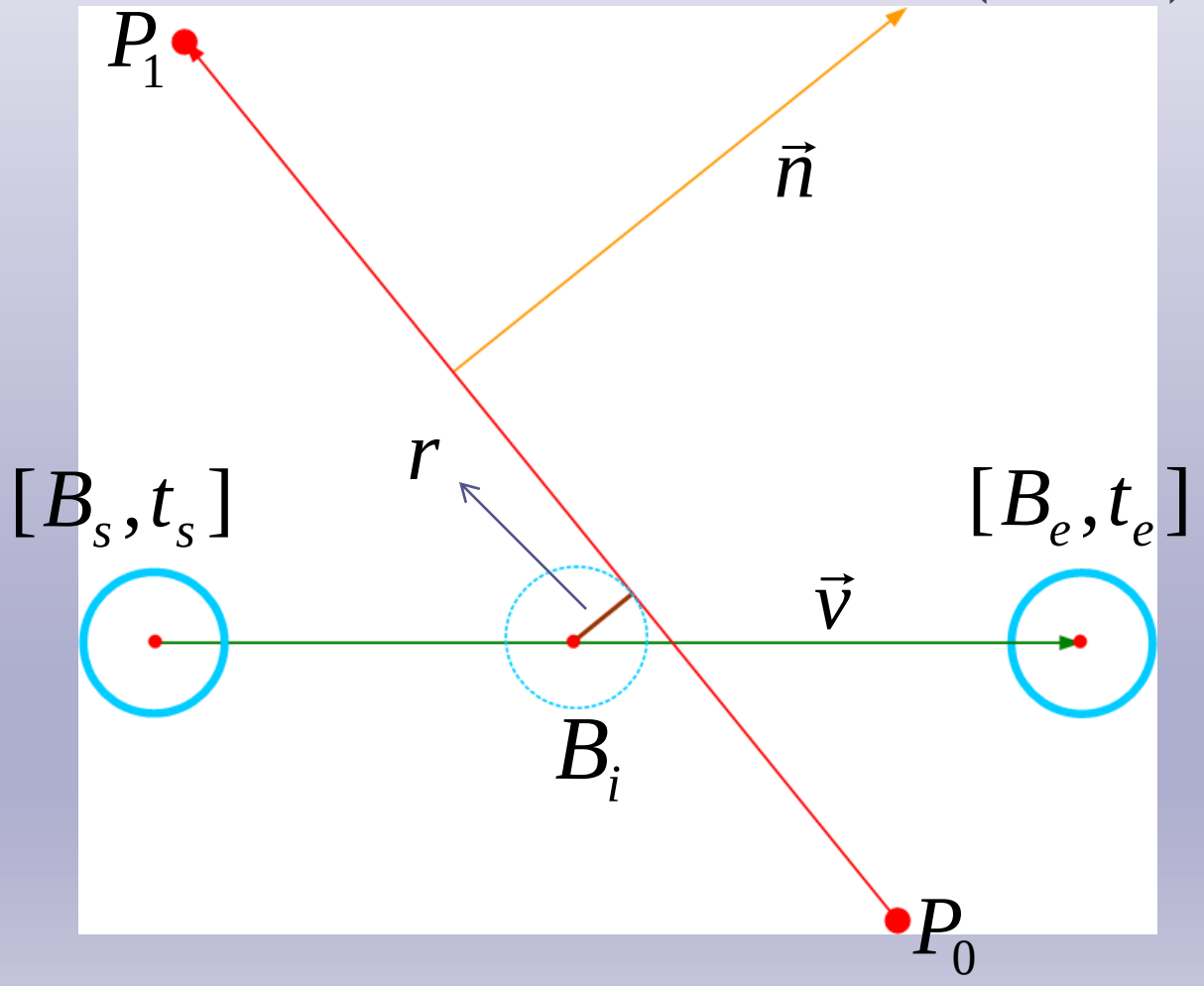
- Pinball modeled by a circle with center and radius r
- Located at center point B_s at beginning of frame
- Moving in direction given by vector $\vec{v}$ and modeled as the following parametric equation

$$\Rightarrow \boldsymbol{B}(t) = \boldsymbol{B}_s + \vec{\boldsymbol{v}}\boldsymbol{t}, \boldsymbol{t} \in [\mathbf{0},\mathbf{1}]$$

- V is the change of position per **frame** $\vec{\boldsymbol{v}} = \overrightarrow{\boldsymbol{B}_s\boldsymbol{B}_e}$

Note: The new position Bs was computed by the Physics system (earlier)

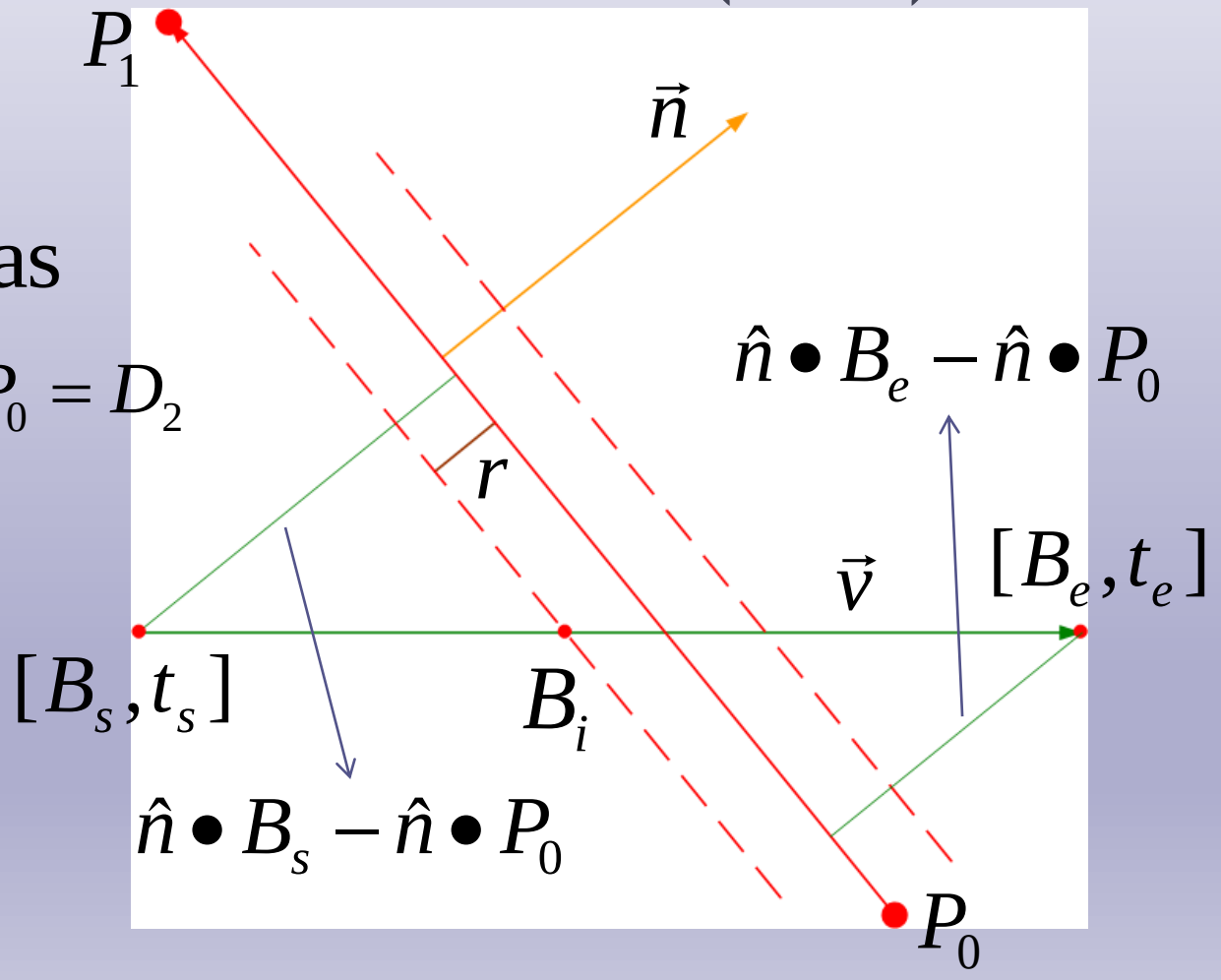
Pinball–Wall Intersection (1/3)



Pinball–Wall Intersection (2/3)

Wall modeled as

$$L : \vec{n} \bullet P - \vec{n} \bullet P_0 = D_2$$



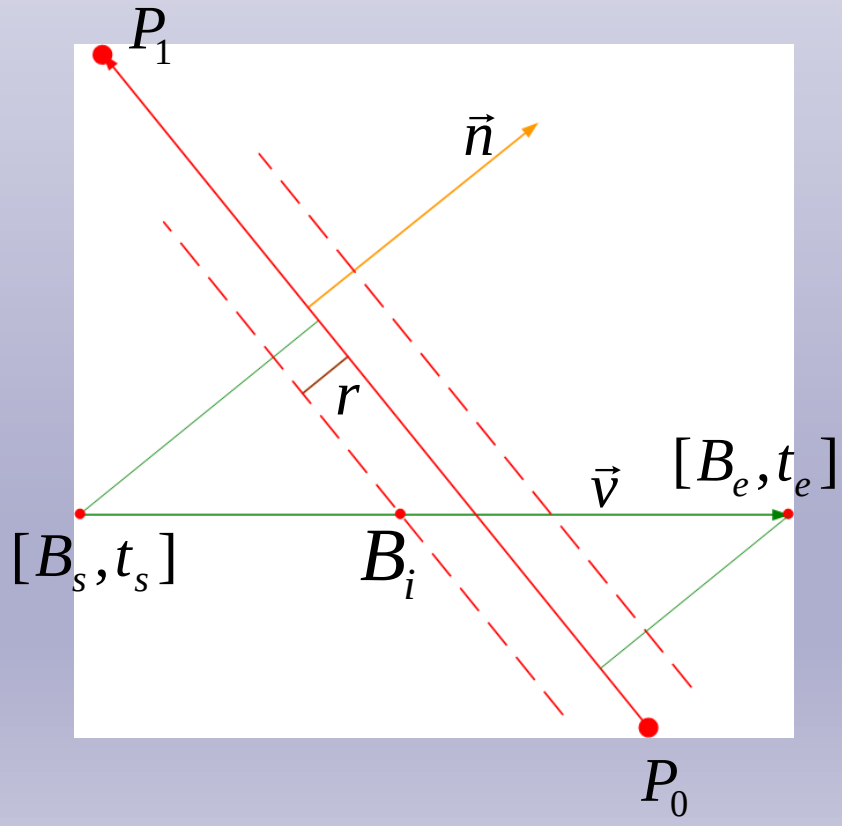
Pinball-Wall Intersection (3/3)

Pinball is inside at t_s and outside at t_e

$$t_i = \frac{\hat{n} \bullet P_0 - \hat{n} \bullet B_s + D_2}{\hat{n} \bullet \vec{v}}$$

where $t_i \in [0,1]$

$$B_i = B_s + \vec{v}\left(\frac{\hat{n} \bullet P_0 - \hat{n} \bullet B_s + D_2}{\hat{n} \bullet \vec{v}}\right)$$

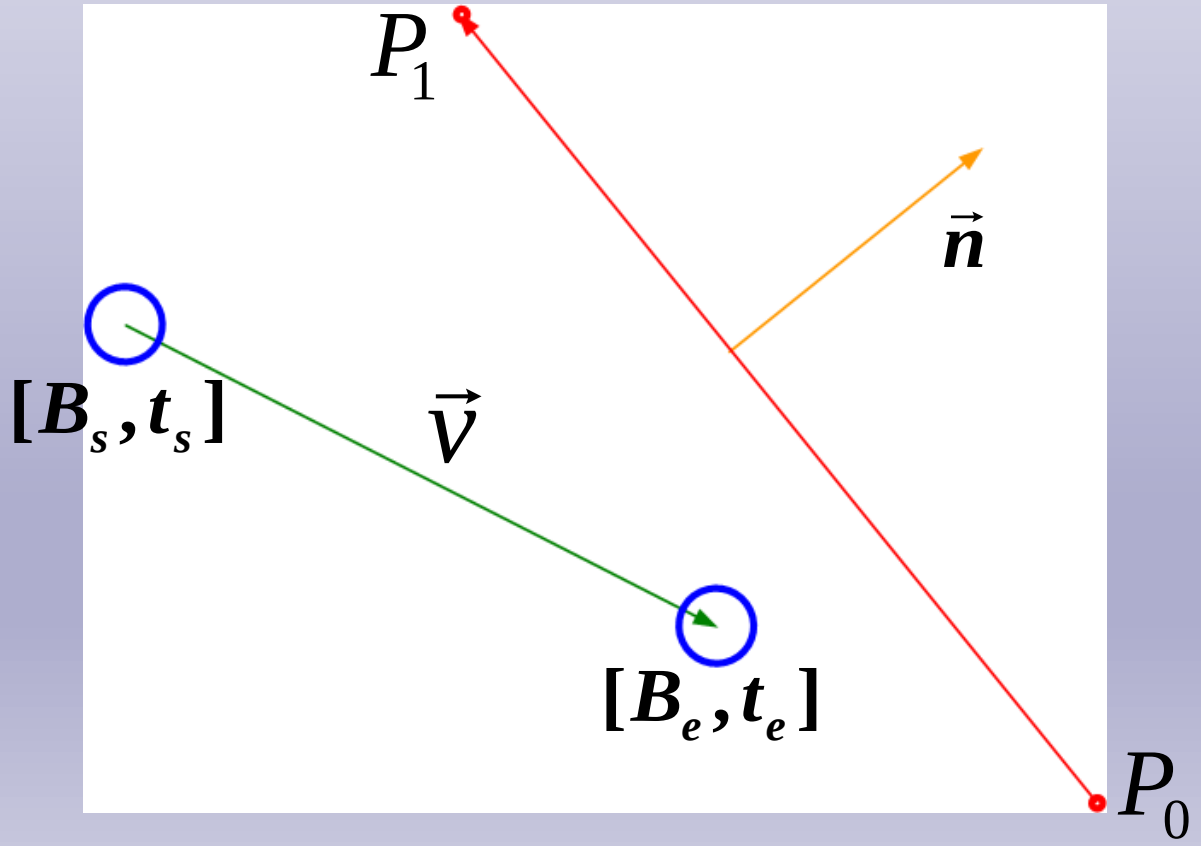


Note: $D_2 = -r$ when B_s is inside
$D_2 = r$ when B_s is outside

Test for Non-Collision (1/4)

Not to be used

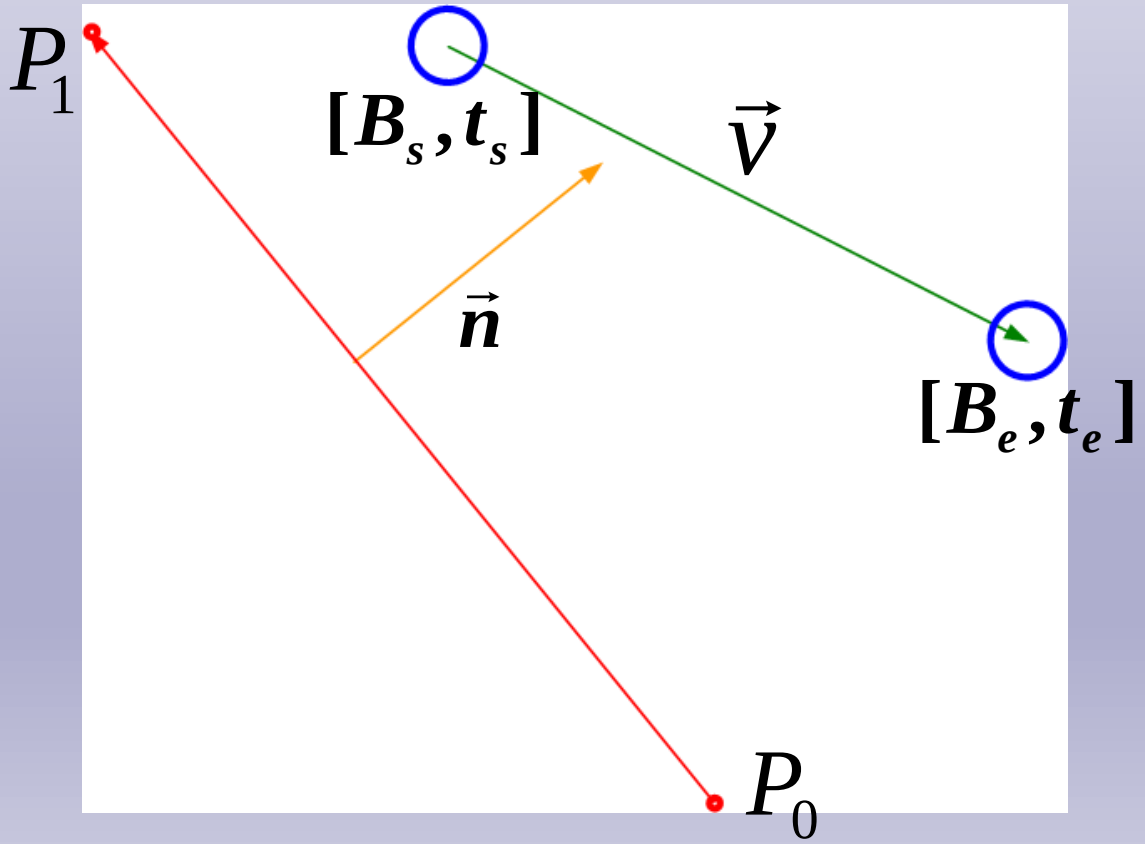
$$(\hat{n} \bullet B_s - \hat{n} \bullet P_0 < -r) \&\& (\hat{n} \bullet B_e - \hat{n} \bullet P_0 < -r)$$



Test for Non-Collision (2/4)

Not to be used

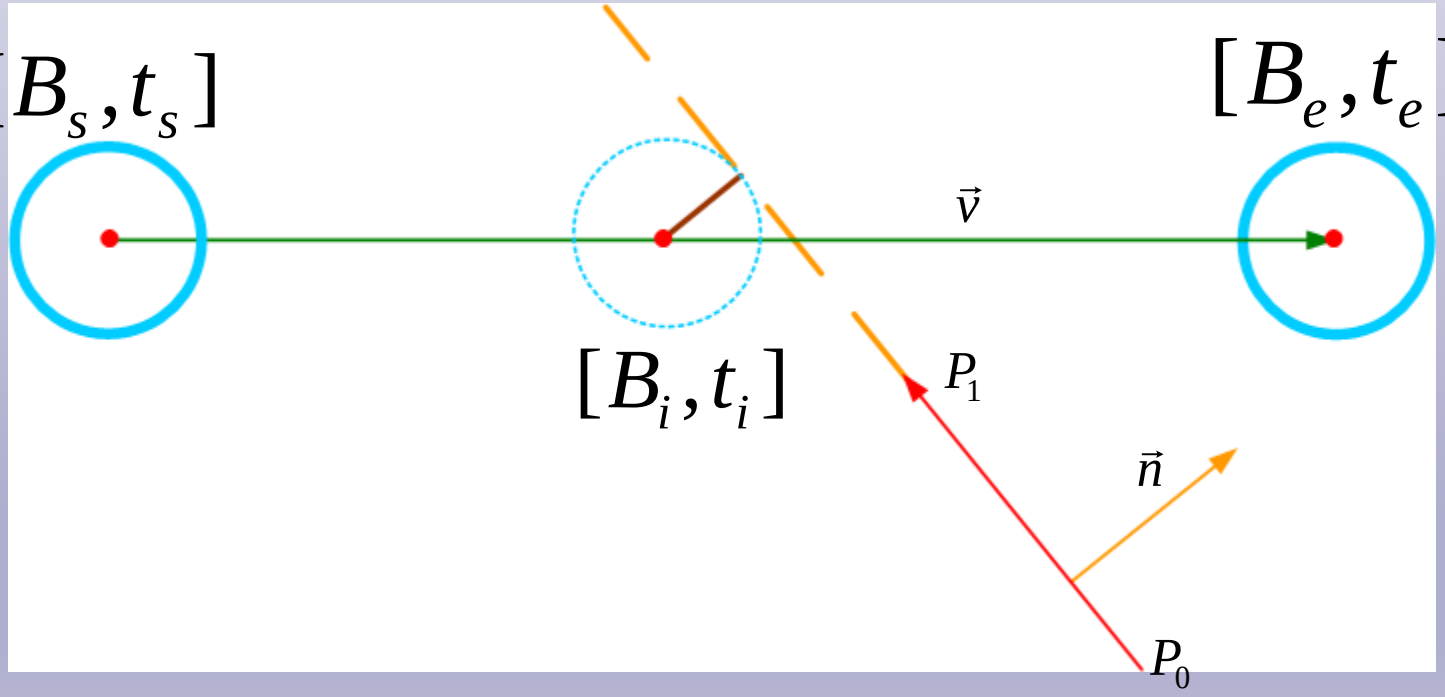
$$(\hat{n} \bullet B_s - \hat{n} \bullet P_0 > r) \&\& (\hat{n} \bullet B_e - \hat{n} \bullet P_0 > r)$$



Test for Non-Collision (3/4)

Not to be used

$$(B_i - P_1) \bullet (P_0 - P_1) < 0$$

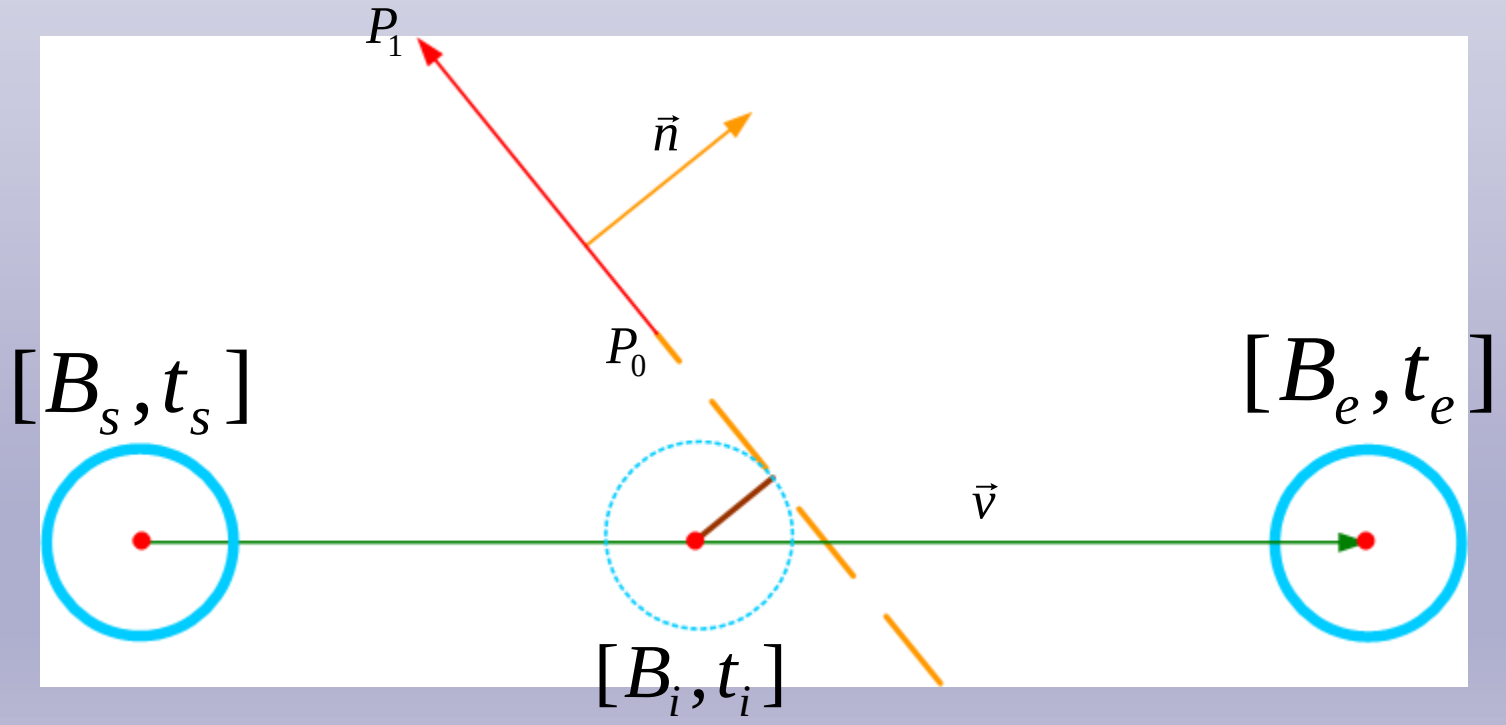


Ball collides with infinite extension of wall… not finite wall!

Test for Non-Collision(4/4)

Not to be used

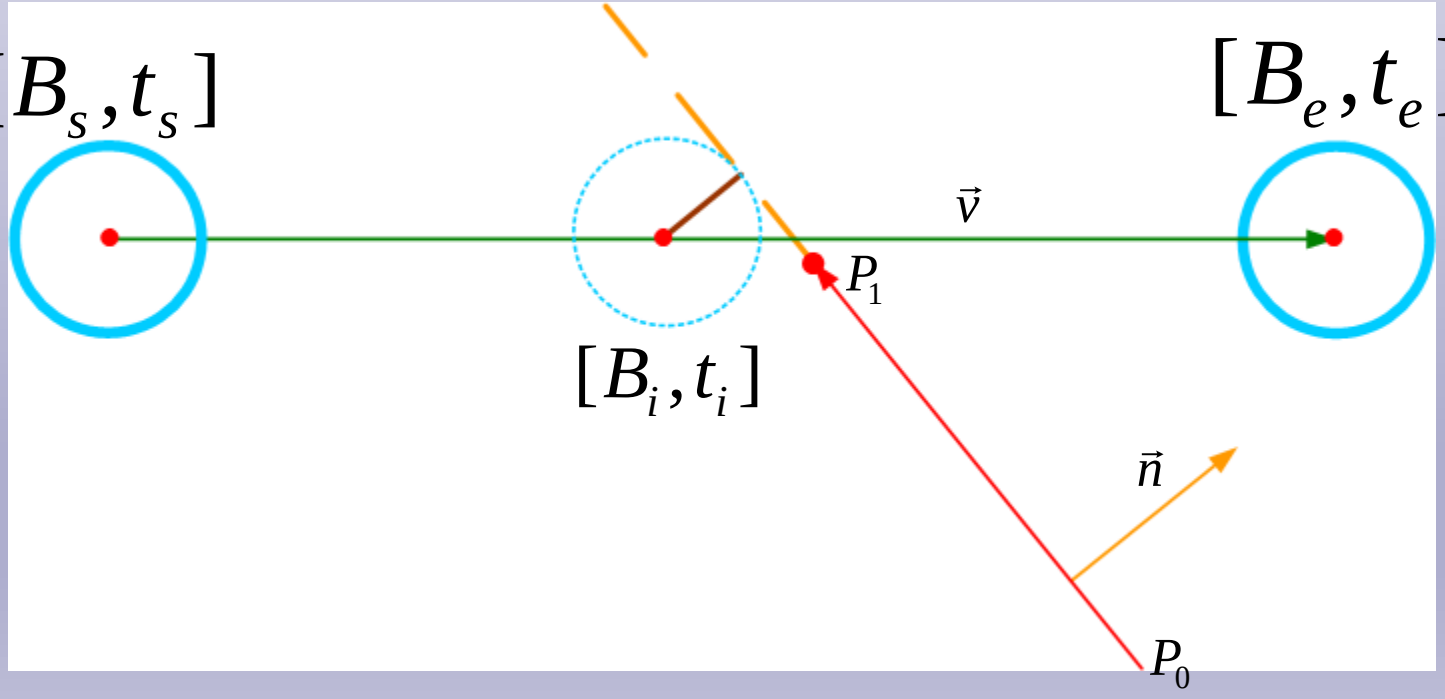
$$(B_i - P_0) \bullet (P_1 - P_0) < 0$$



Ball collides with infinite extension of wall… not finite wall!

But! We have a Problem

The intersection point B_i is not on the line, but the ball collides with the wall.



Steps

- Check for trivial rejection
  - B_s, B_e inside or both outside
  - Going from inside to outside and the collision type of the line segment is outside, and vice versa.
- Calculate the point and time of intersection
  - Check if the time is between t_s and t_e
- Check if the point of intersection is on the line segment

Not to be used
We're going to see a better algorithm with more steps