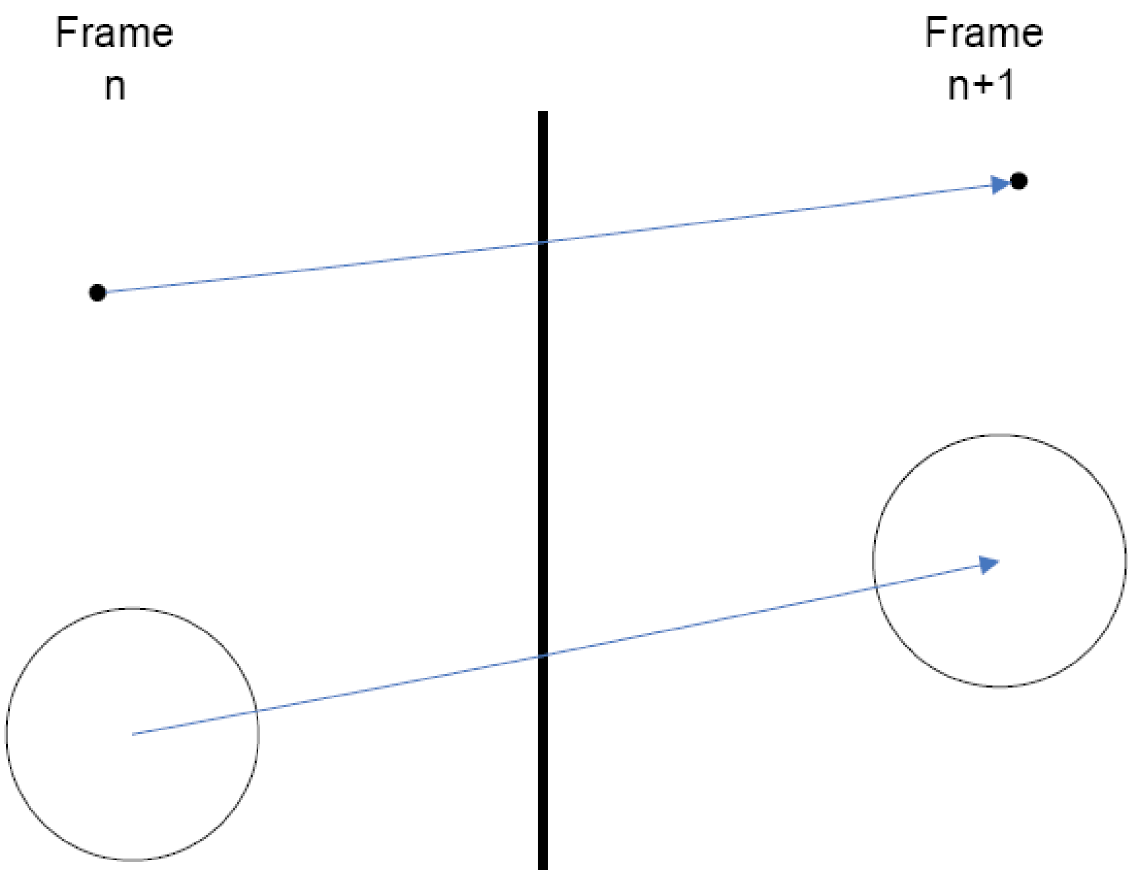
Lecture 16b | Normal Line Equation

CS230
Game
Implementation
Techniques

1. Time Based Collision

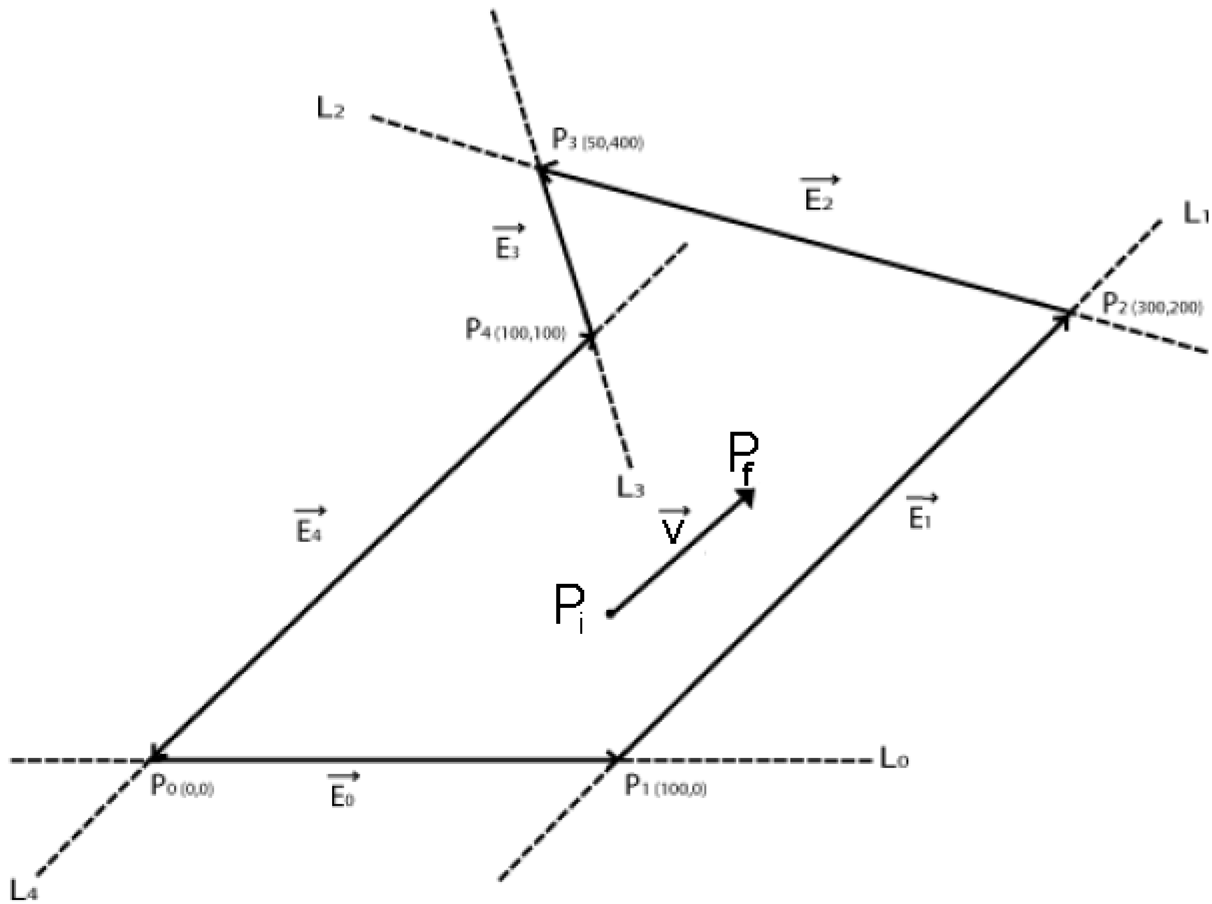
1.1. Point-Line

- The circle or point is moving and the line segment is static.
- Checking for collision at the previous and current positions is not enough.
- The circle (or the point) and the line can be not colliding with the line at frame n and frame n+1, but in reality they might have collided.



- As you can see in the figure above, the circle and the point aren't colliding with the line at frame n, and aren't colliding with it at frame n+1, but obviously, they are colliding with it sometime along the path.
- We need to interpolate along the path in order to find the exact time where the collision occurred.

1.2. Intersection of a ray with a line segment



Edge	Vector	Outward Normal
0	$\overrightarrow{E_0}$ = (100 ; 0)	$\overrightarrow{N_0}$ = (0 ; -100)
1	$\overrightarrow{E_1}$ = (200 ; 200)	$\overrightarrow{N_1}$ = (200 ; -200)
2	$\overrightarrow{E_2}$ = (-250 ; 200)	$\overrightarrow{N_2}$ = (200 ; 250)
3	$\overrightarrow{E_3}$ = (50 ; -300)	$\overrightarrow{N_3}$ = (-300 ; -50)
4	$\overrightarrow{E_4}$ = (-100 ; -100)	$\overrightarrow{N_0}$ = (-100 ; 100)

Infinite Line	Equation
0	(0 ; -100).(x ; y) = 0
1	(200 ; -200).(x ; y) = 20000
2	(200 ; 250).(x ; y) = 110000
3	(-300 ; -50).(x ; y) = -350000
4	(-100 ; 100).(x ; y) = 0

- Consider a point to be located at P_i inside the chamber and moving with towards P_f
- This means that its velocity is $\overrightarrow{P_iP_f} = \vec{V}$
- Let $\vec{P}$ be the position vector of point P (The moving point). Then the equation of the ray is: $\vec{P} = \overrightarrow{P_iP_f}*t + P_i = \vec{V}*t + P_i$
- At which time t_n, does the ray hit the line L given by: $\vec{N}.P = D$?
- Assume P_n is the spot where the ray intersects the line L
- This means that the ray intersects with the line at point P_n at time t_n
- This means that the point P_n belongs to both the ray and the line
- Thus, P_n in $\vec{N}.P_n = D$ is the same as $\vec{P}_n = \vec{V}*t_n + P_i$ (It's the same point, the point of intersection)
- By replacing the 2nd equation in the first one we get:

$$\vec{N}.(\vec{V}*t_n + P_i) = D$$

$$\vec{N}.\vec{V}*t_n + \vec{N}.P_i = D$$

$$\vec{N}.\vec{V}*t_n = D - \vec{N}.P_i$$

$$t_n = \frac{D - \vec{N}.P_i}{\vec{N}.\vec{V}}$$

- This means that $\vec{N}.\vec{V} \neq 0$

  If $\vec{N}.\vec{V} = 0$ the ray is parallel to the line and will never hit it.

- When there is a hit, we find the coordinates of $\vec{P}_n = \vec{V}*t_n + P_i$
- If t_n is less than 0 or greater than 1, than there is no collision.

DigiPen INSTITUTE OF TECHNOLOGY

Collision Exercise:

Given the following line segment L[P0(6 ; 0), P1(10 ; 7)], representing a wall in your game, we need to check if the following moving point objects cases may collide with L in the current frame?
Each point object is moving from $\mathbf{B_s}$ to $\mathbf{B_e}$.

To answer these questions, you need to apply the equations found in the "***Lecture 16 - Normal Line Equation - Animated Point To Line.pdf***" file, between slides 18 and 27.

Please follow these steps:
- We'll skip the first 3 non-collision tests: (1/5), (2/5) and (3/5)
- Compute ti
- If (0 <= ti <= 1), compute Bi
- Test for non-collision (4/5) and (5/5). Or you can combine both test into one.

Case 1:

$\mathbf{B_s}$ **= (5 ; 2) ,** $\mathbf{B_e}$ **= (7 ; 3)**

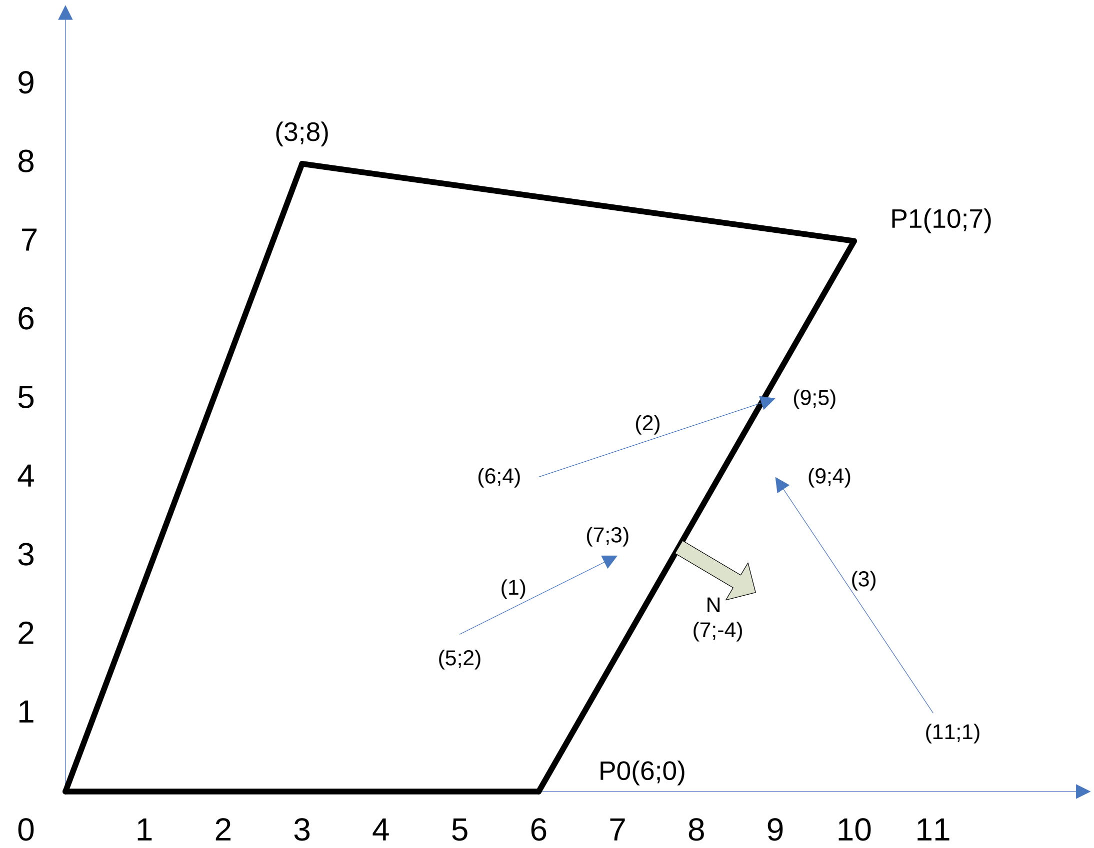
Case 2:

$\mathbf{B_s}$ **= (6 ; 4) ,** $\mathbf{B_e}$ **= (9 ; 5)**

Case 3:

$\mathbf{B_s}$ **= (11; 1) ,** $\mathbf{B_e}$ **= (9 ; 4)**

- Examples: Checking for collision with line P0P1



- We need to find normal line equation of P0P1
  - $\overrightarrow{P_0P_1} = P_1 - P_0$ = (10 – 6 ; 7 – 0) = (4 ; 7)
  - $\vec{N} = (\overrightarrow{P_0P_1}.y\ ;\ -\overrightarrow{P_0P_1}.x)$ = (7 ; -4)
  - Finding the constant D of the equation:
    $\vec{N}.P_1$ = (7 ; -4).(10 ; 7) = 70 – 28 = 42 = D
    (Any point of the line works. We could have used P_0):
    $\vec{N}.P_0$ = (7 ; -4).(6 ; 0) = 42 – 0 = 42 = D
  - Normal line equation:
    - P(x;y) is any point on P0P1
    - $\vec{N}$.P = D
    - (7 ; -4).(x ; y) = 42

- Case (1):
  - B_s = (5 ; 2)
  - B_e = (7 ; 3)
  - $\vec{V} = B_e - B_s$ = (7 ; 3) – (5 ; 2) = (2 ; 1)
  - We should find the collision time t_i using the equation $t_i = \frac{D - \vec{N}.B_s}{\vec{N}.\vec{V}}$
  - $t_i = \frac{42-(7\,;-4).(5\,;2)}{(7\,;-4).(2\,;1)} = \frac{42-35+8}{14-4} = \frac{15}{10} = 1.5$
  - t_i is greater than 1 → No collision

- Case (2):
  - B_s = (6 ; 4)
  - B_e = (9 ; 5)
  - $\vec{V} = B_e - B_s$ = (9 ; 5) – (6 ; 4) = (3 ; 1)
  - We should find the collision time t_i using the equation $t_i = \frac{D - \vec{N}.B_s}{\vec{N}.\vec{V}}$
  - $t_i = \frac{42-(7\,;-4).(6\,;4)}{(7\,;-4).(3\,;1)} = \frac{42-42+16}{21-4} = \frac{16}{17} = 0.94$
  - $0 \le t_i \le 1$ → then the ray collides with the line L at time t_i = 0.94
  - To get the exact point of intersection, we should replace t_i in the ray equation.

    $\vec{B}_i = \vec{V} * t_i + B_s$ = (3 ; 1)*0.94 + (6 ; 4) = (2.82 ; 0.94) + (6 ; 4) = (8.82 ; 4.94)

    To verify that this point belongs to the line L, its coordinates should satisfy the normal line equation of L:
    $\vec{N}$.P = D

    (7 ; -4).(8.82 ; 4.94) = 42
    61.74 – 19.76 ≅ 42 (The lost precision is due to the fact that we're using 2 decimal digits)
    Check if outside boundaries (Non-Collision tests 4 and 5 combined):

    $(B_i - P_0) . (B_i - P_1)$ =

    [(8.82 ; 4.94) - (6 ; 4)] . [(8.82 ; 4.94) - (9 ; 5)] = [2.82 ; 0.94] . [-0.18 ; -0.06] < 0

    ⇨ B_i is inside

Case (3)

- B_s = (11 ; 1)
- B_e = (9 ; 4)
- $\vec{V}$ = B_e - B_s = (9 ; 4) – (11 ; 1) = (-2 ; 3)
- We should find the collision time t_i using the equation $t_i = \frac{\mathrm{D} - \vec{N}.B_S}{\vec{N}.\vec{V}}$
- $t_i = \frac{42-(7\,;-4).(11\,;1)}{(7\,;-4).(-2\,;3)} = \frac{42-77+4}{-14-12} = \frac{-31}{-26} = 1.19$
- t_i is greater than 1 → No collision