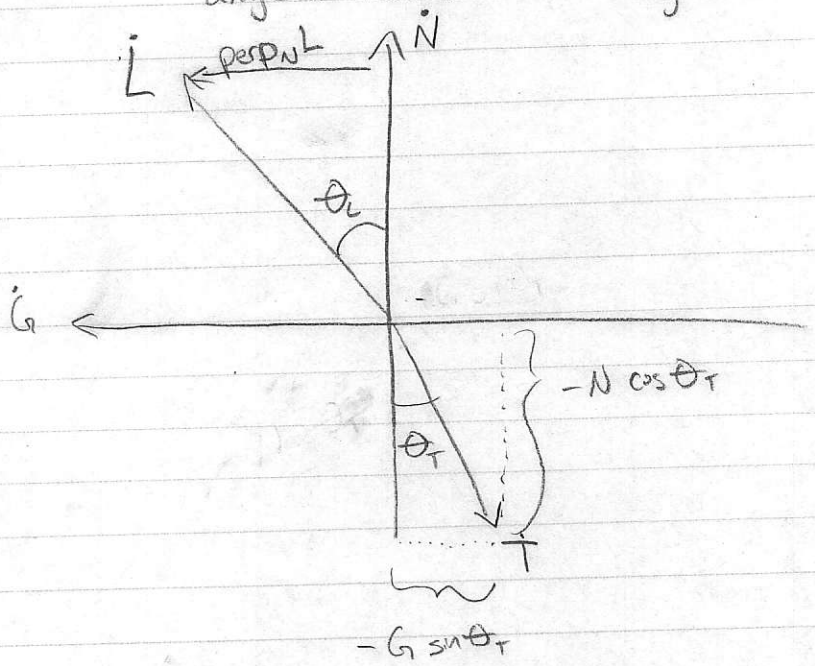
Refraction

- Index of refraction

Snell's Law

$$n_L \sin\theta_L = n_T \sin\theta_T$$

θ_L: angle of incidence; θ_T: angle of transmission



Break T into 2 components, 1 parallel to N, the other perpendicular

Parallel: $-N\cos\theta_T$

Perpendicular: $-G\sin\theta_T$

$\|\text{perp}_N L\| = \sin\theta_L$, so $G = \frac{\text{perp}_N L}{\sin\theta_L} = \frac{L-(N\cdot L)N}{\sin\theta_L}$

(since L is normalized)

$$T = -N\cos\theta_T + -G\sin\theta_T$$
$$= -N\cos\theta_T - \frac{\sin\theta_T}{\sin\theta_L}(L-(N\cdot L)N)$$
$$= -N\cos\theta_T - \frac{n_L}{n_T}(L-(N\cdot L)N)$$

$$\cos\theta_T = \sqrt{1-\sin^2\theta_T}$$

$$T = -N\sqrt{1-\sin^2\theta_T} - \frac{n_L}{n_T}\left(L-(N\cdot L)N\right)$$

$$\sin\theta_T = \left(n_L/n_T \sin\theta_L\right)$$

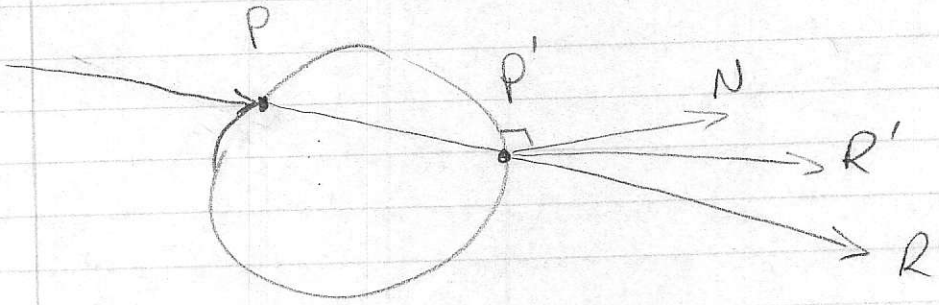
$$= -N\sqrt{1-\frac{n_L^2}{n_T^2}\sin^2\theta_L} - \frac{n_L}{n_T}\left(L-(N\cdot L)N\right)$$

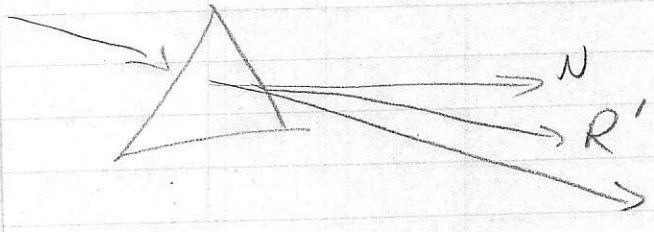
$$\sin^2\theta_L = 1-\cos^2\theta_L = 1-(N\cdot L)^2$$

$$= \left(\frac{n_L}{n_T}(N\cdot L) - \sqrt{1-\frac{n_L^2}{n_T^2}\left(1-(N\cdot L)^2\right)}\right)N - \frac{n_L}{n_T}L$$

- Be careful that $\sin\theta_L \leq n_T/n_L$, when this is the case light is not refracted outside the object.

Easy Way (not full bonus marks)

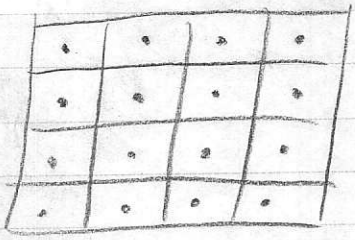




- Be careful not to hit originating object!

Anti-Aliasing

→ Stop the jaggies



- 2 behaviors to help
  1. more rays
  2. random jitter

1 pixel

f	b	j
c	a	d
h	e	i

→ send 9 rays into the pixel
→ perhaps re-use results from neighbor

$a = \text{topleft} + \text{downvec} * \text{down} + \text{overvec} * \text{over}$

random $\Delta\text{down} = \frac{1}{\text{height}-1} * \text{downvec}$

$\Delta\text{right} = \frac{1}{\text{width}-1} * \text{overvec}$

$a = \text{topleft} + \text{downvec} * \text{down} + \text{overvec} * \text{over}$
$+ \frac{1}{3}(\text{random}(-0.5, 0.5) * \Delta\text{down})$
$+ \frac{1}{3}(\text{random}(-0.5, 0.5) * \Delta\text{right})$

$f = \text{topleft} + \text{downvec} * \text{down} + \text{overvec} * \text{over}$
$- \frac{1}{3}\Delta\text{right} - \frac{1}{3}\Delta\text{down}$
$+ \frac{1}{3}(\text{random}(-0.5, 0.5) * \Delta\text{down})$
$+ \frac{1}{3}(\text{random}(-0.5, 0.5) * \Delta\text{right})$

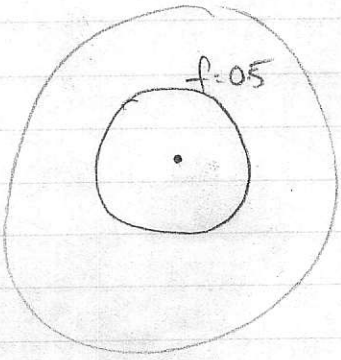
rand → stdlib
rand() / RAND_MAX

→ average or weighted average

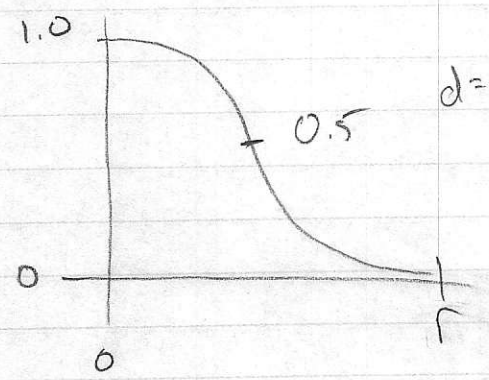
$$(9 \cdot a + 3(b+c+d+e) + (f+g+h+i)) / 28$$

Implicits

- aka blobs, meta-balls



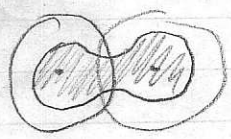
$$F(x,y,z) = 1 - \frac{4}{9}\frac{d^6}{r^6} + \frac{17}{9}\frac{d^4}{r^4} - \frac{22}{9}\frac{d^2}{r^2}$$



$$d = \sqrt{(x-c_x)^2 + (y-c_y)^2 + (z-c_z)^2}$$

Trick, fields from neighboring implicits add up

$$f(x,y,z) = \sum_{i=0}^{n} F_i(x,y,z)$$



→ other primitives possible including lines, polygons etc

→ simple way
- test field value at every point along ever ray

→ faster way
- do sphere intersection on point
- if hit to field testing until (if) you hit field value