

# Midterm review (brief)

C++ :

- What are the "big three"  
(destructors copy const.,  
assignment operator =)

- know how operators  
work

- know how iterators  
work

Bij-oh:

- know formal definitions

$O(T(n))$

$\Theta(T(n))$

$\Omega(T(n))$

- Know defs, meaning

- practical implications

(dis-compare  $\xi$  constant  
also)

# Binary Trees

- know definitions,  
properties

e.g., what does it mean  
for a binary tree to be  
complete?

# Heaps

- how does heapsort work?

- how fast is it  
( $\Theta(n \log n)$ )?

# AVL Trees

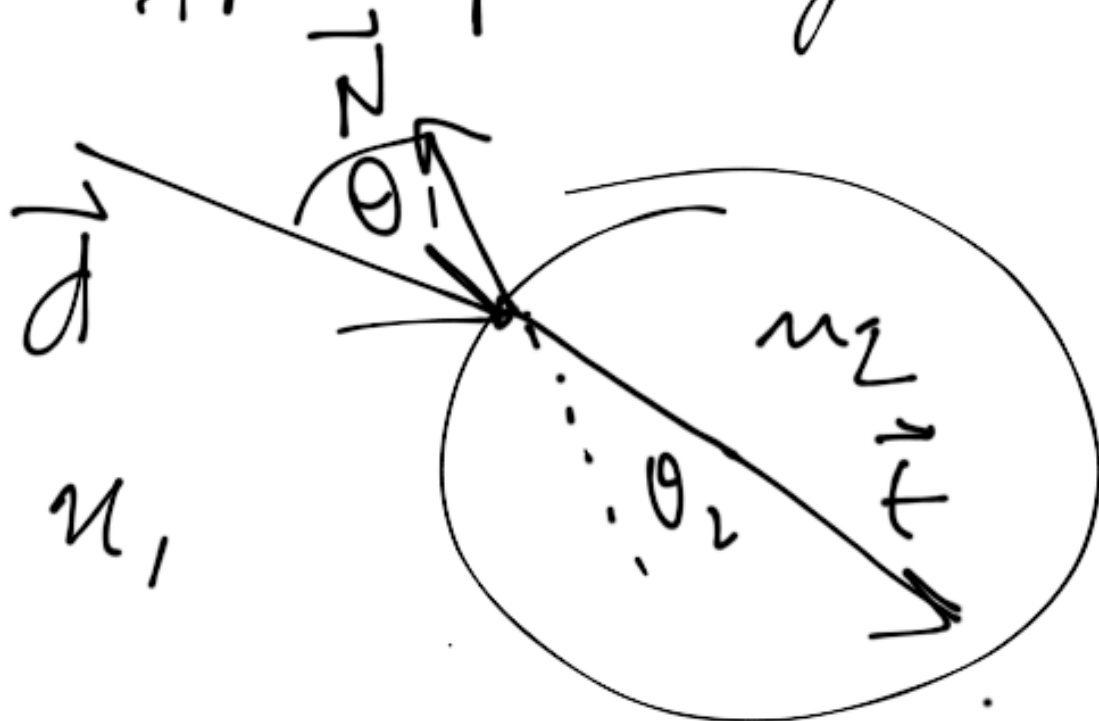
— Know how to

insert, delete,

do rotations

AS62

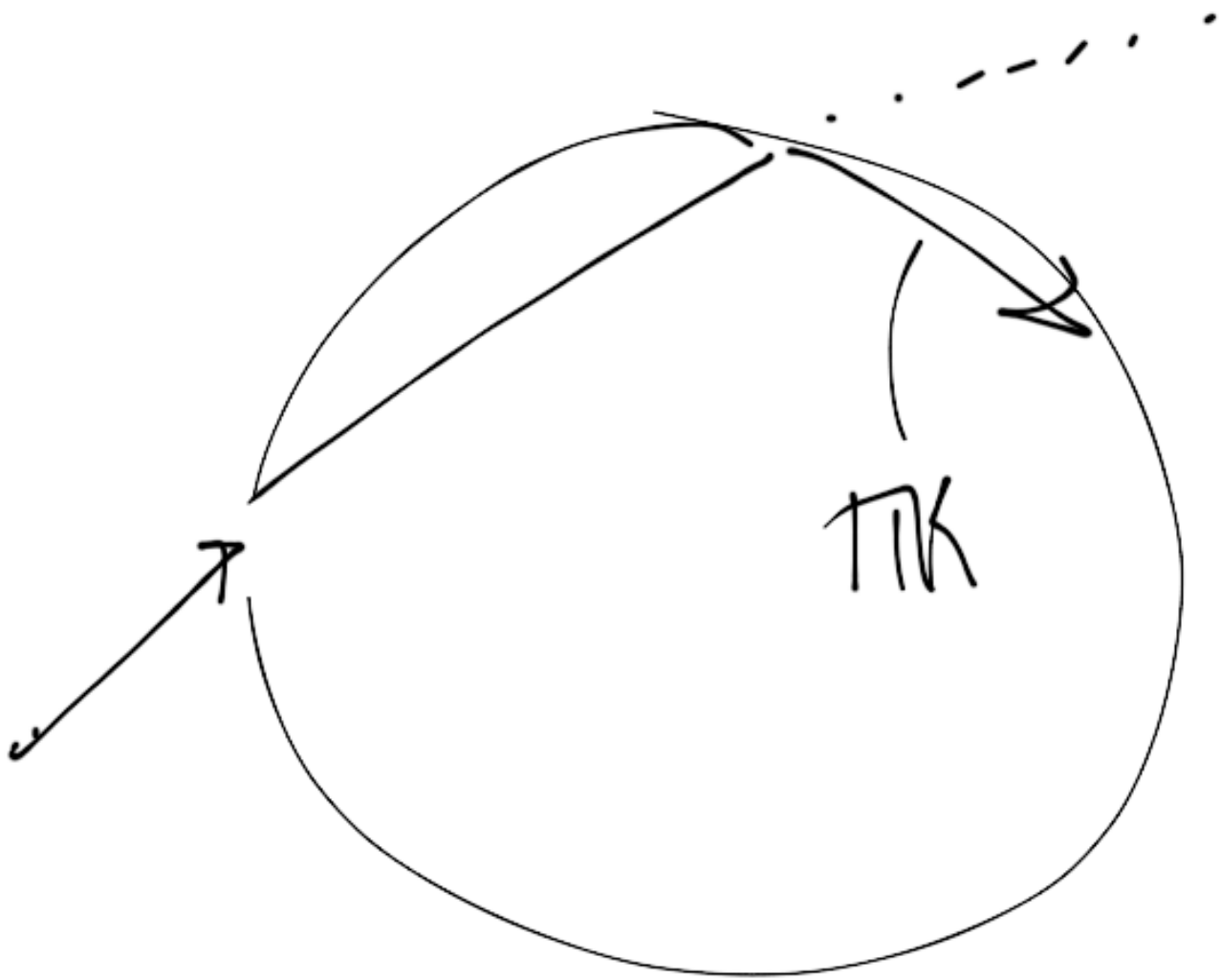
- transparency



$$n_1 \sin \theta_1 = n_2 \sin \theta_2$$

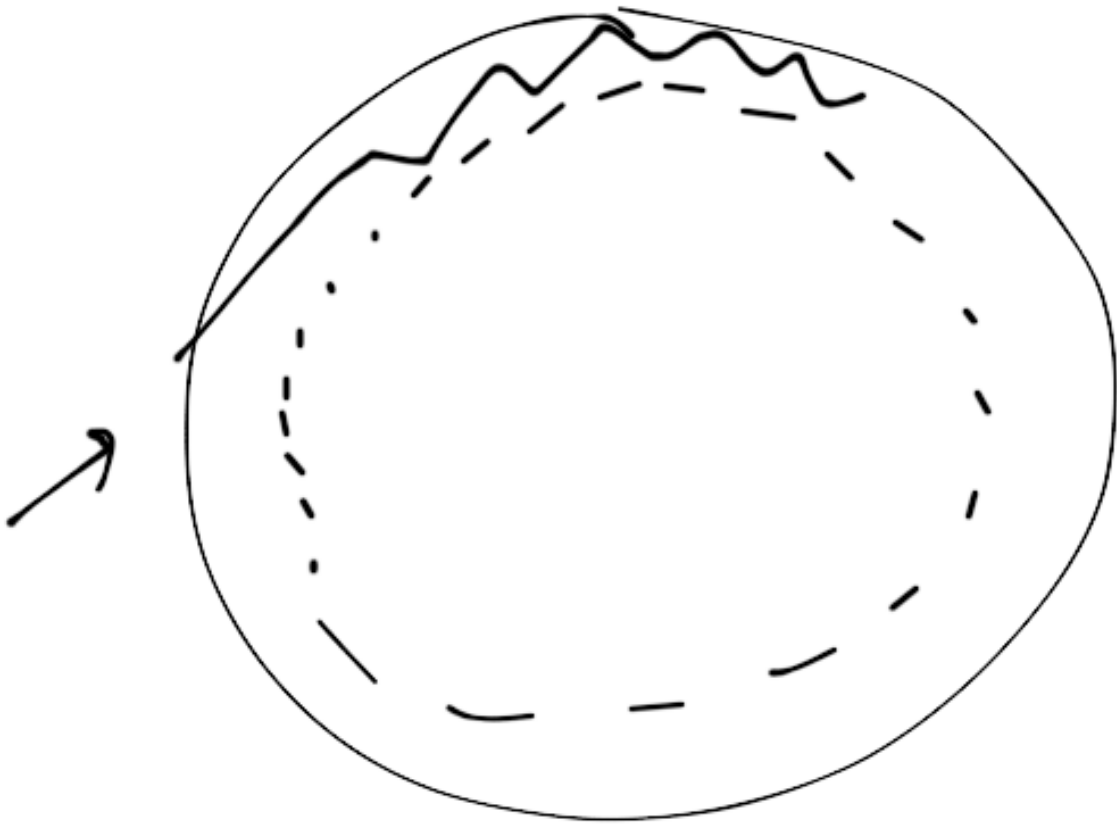
$$\vec{t} = \frac{n_1}{n_2} (\vec{d} - \vec{N} (\vec{d} \cdot \vec{N})) -$$

$$\vec{N} \sqrt{1 - \left(\frac{n_1}{n_2}\right)^2 (1 - (\vec{d} \cdot \vec{N})^2)}$$



if stuff is  $\sqrt{\quad}$  is  $-V_e$ ,

TIR: Total internal  
reflection



in main() :

ray = new ray\_t (  $\vec{p}$ ,  $\vec{d}$  )

ray → trace (  $\emptyset$  )

↑  
bounce



in ray\_t::trace()

if (bounce > 10) return;

if (sphere & !transparent)

r = d.reflect(N)

↳ ray direction

if (|d · N| < 0) (≈ vec\_t)

t = d.refract(N, n<sub>2</sub>)

else

t = d.refract(-N, 1/n<sub>2</sub>)

transmission = new

ray, t (+)

color = transmission \* color (t + 60% a)

$$\text{color} = ((1-\alpha)\text{color} + (\alpha) \text{t color})$$

linear interpolation

between colors (blending)

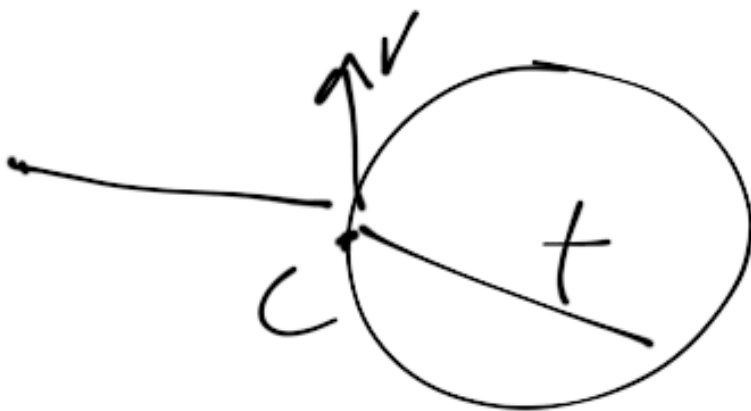
Schlische:

blendebetten:

color,

t-color. (transmitter)

ref color



AS6 9

azi.  $\rightarrow$   $\theta$  elev.

Photon mapping

point light

1. shoot photons in all (random) directions

2. ray trace



"caustic":

like a specular reflector  
highlight

AS6 4:

just "phacet" photos,

add to list

(std::vector)

AS6 6:

insert photos from

list into kd-tree,

ray trace