# Ray/plane intersection (plane_hits())

Monday, September 13, 2010
8:58 AM

Assignment #1: test plane_hits() function

Due Next week (Monday; has to be done between labs 4 & 5)

lab 5 depends on plane_hits()

ray-plane intersection (p. 72 in notes)

Monday, September 13, 2010
9:09 AM

from last time:

pln->ndotq = vec_dot(pln->normal, pln->point)

what is this? this is D in the plane's equation

both of these are in input file

plane is defined as $Ax + By + Cz + D = 0$

for any point $(x, y, z)$

plane is defined by $(A, B, C)$ D

$(A, B, C)$ = plane normal

pln->normal = $(A, B, C)$

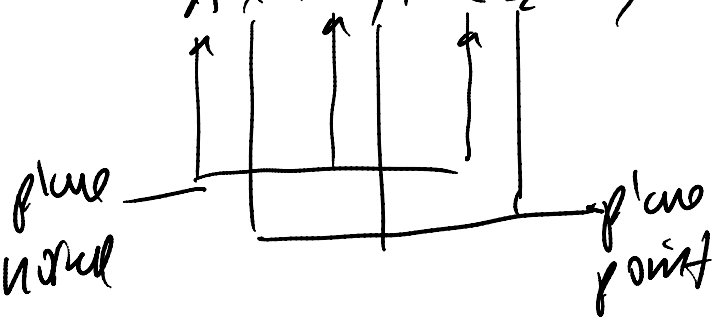
from file

Now solve for D: take any known point on the plane $(x, y, z)$, plug into plane equation, solve for D

pln->point = (x, y, z)

from file

$Ax + By + Cz + D = 0$




$$D = -(Ax + By + Cz)$$

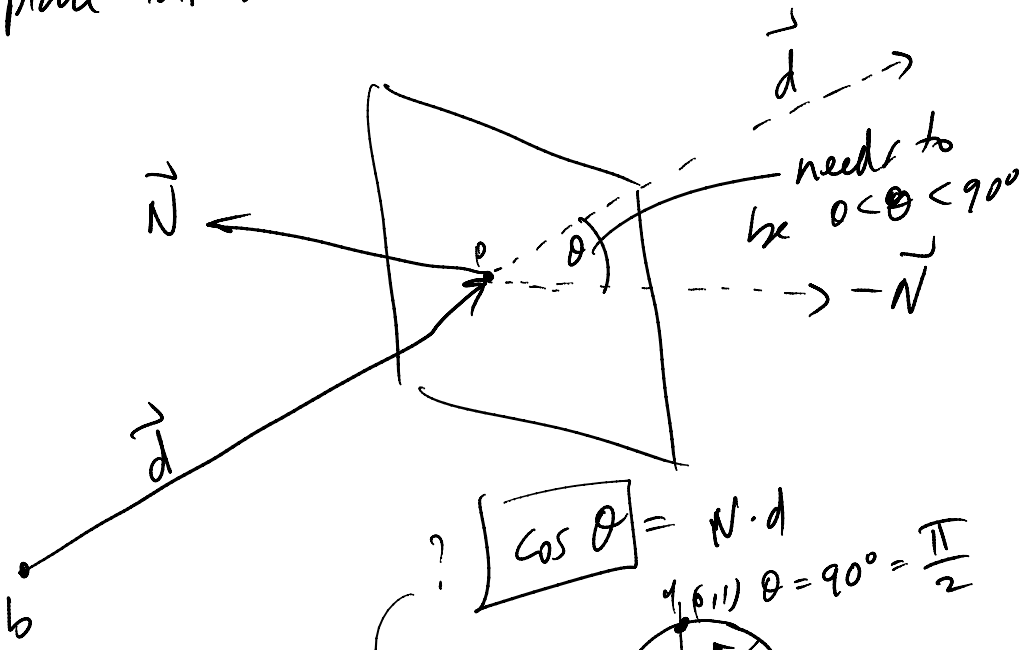
$$\begin{pmatrix} A \\ B \\ C \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$= -(N \cdot P)$$

pln->ndotq = vec_dot(pln->normal, pln->point)

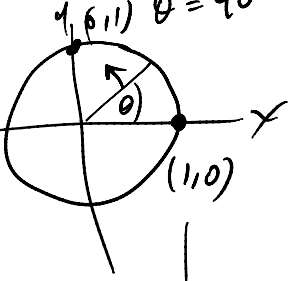
Monday, September 13, 2010
9:17 AM

ray / plane intersection




b: base
d: dir

? $\boxed{\cos\theta = N \cdot d}$




$(\cos\theta, \sin\theta)$

$\theta = 0$

1. $N \cdot d$

↑ ↑

plane normal, ray dir, normalized

if $(N \cdot d == 0.0)$

$\theta = 90°$, no intersection

return (-1)

fabs($N \cdot d$) < precision

|

float absolute val

$|N \cdot d|$

2. ($N \cdot d$ is not 0, we have intersection.

calculate distance from ray's base to plane) $(D - (N \cdot b))$

calculate $\vec{N} \cdot \vec{b}$, call this $t$

($\vec{N}$: plane normal; $\vec{b}$: ray base; vec_dot(N, b); just a scalar, a float)

3. scale $t$ by $N \cdot d$

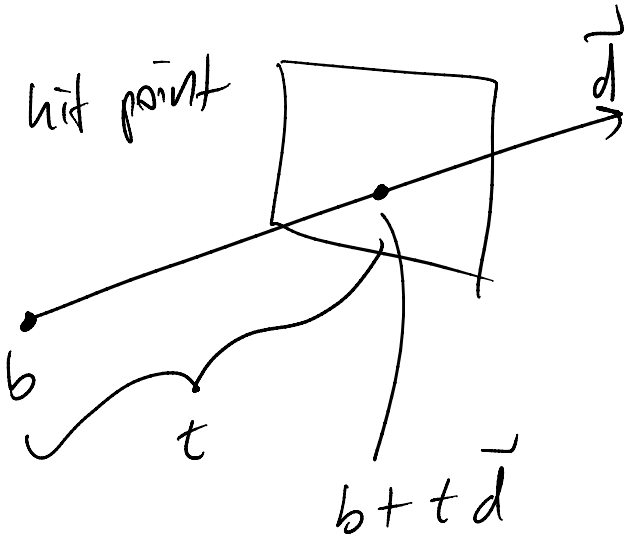
$$t = (D - (N \cdot b)) / (N \cdot d)$$

(plane->ndotb - vec_dot(plane->normal, base)) / vec_dot(plane->normal, dir)

ray base, ray dir

Come in as arguments to plane_hits()

4. calculate hit point




hit point = base + vec_scale(t, d)

(vec_t)

↑

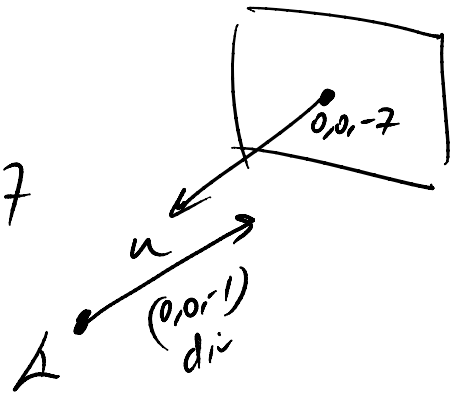
vector addition?

vec_sum(a, vec_scale(t, d))

(check syntax — don't just type this in

Numerical example
(or) manual pen-on-paper program trace
[another way to debug code]

given: plane n: 0 0 1
p: 0 0 −7




ray pt: 4, 3, 5
dir: 0, 0, −1

1. vec_unit(dir, dir) // normalize plane dir

$$\sqrt{0^2+0^2+1^2} = \sqrt{1^2} = 1$$

$$dir = \frac{1}{1}\begin{pmatrix}0\\0\\-1\end{pmatrix} = \begin{pmatrix}0\\0\\-1\end{pmatrix}$$

2. n dot d $= \begin{pmatrix}0\\0\\1\end{pmatrix} \cdot \begin{pmatrix}0\\0\\-1\end{pmatrix} = \sum\begin{pmatrix}0\\0\\-1\end{pmatrix} = -1$

3. n dot b $= \begin{pmatrix}0\\0\\1\end{pmatrix} \cdot \begin{pmatrix}4\\3\\5\end{pmatrix} = \sum\begin{pmatrix}0\\0\\5\end{pmatrix} = 5$

4. $t = \dfrac{\text{pln2 ndot } g - \text{ndot } b}{\text{ndot } d} = \dfrac{-7-5}{-1} = \dfrac{-12}{-1} = 12$

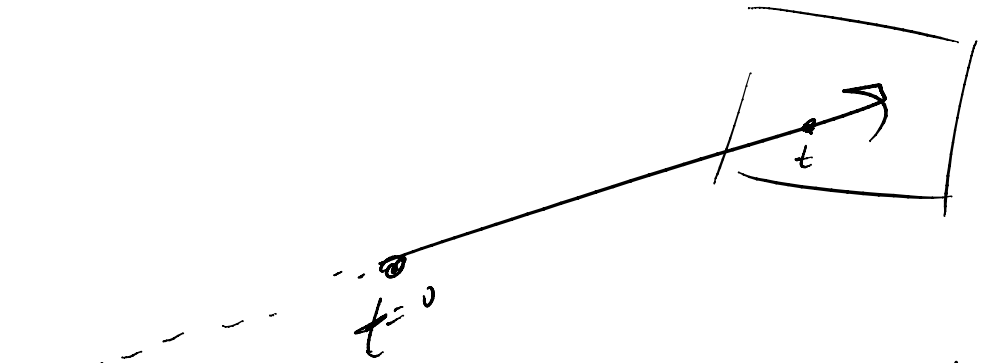
pln → ndot g = $n \cdot p = \begin{pmatrix}0\\0\\1\end{pmatrix} \cdot \begin{pmatrix}0\\0\\-7\end{pmatrix} = \sum \begin{pmatrix}0\\0\\-7\end{pmatrix} = -7$

if $(t < 0)$ return $(-1)$

5. hit point

base + t dir

$$\begin{pmatrix}4\\3\\5\end{pmatrix} + 12\begin{pmatrix}0\\0\\-1\end{pmatrix} = \begin{pmatrix}4\\3\\5-12\end{pmatrix} = \begin{pmatrix}4\\3\\-7\end{pmatrix}$$




6- store hit point in obj → last_hit

if ( obj → last_hit[2] > 0.0 )

return (-1)

return (-1)

int. point in front of image plane