Tristan Rice (25886145, q7w9a)
CS314 - H3
2016-03-08

This assignment was done in R. Original file: https://fn.lc/s/h3.R

```
> B = c(2, 3, 0)
> C = c(8, 12, 0)
> D = c(6, 9, 0) # not a vertex
> Na = c(-0.5, -0.3, 0)
> Nc = c(0.9, 0.5, 0)
> Nf = c(-1, -1, 0)
> Ng = c(1, -1, 0)
> #eye point
> E = c(1,-2,0)
> #light position
> L = c(-1,0,0)
> #ambient light color
> Ia = c(0.5, 0.1, 0.5)
> #light color
> Il = c(0.9, 1, 1)
> #diffuse material color
> kd = c(.3, .8, .2)
> #ambient material color
> ka = c(.5, .2, .5)
> #specular material color
> ks = c(0.5, 1, 1)
> #shininess exponent
> kse = 10
>
> normalize = function(a) a/sqrt(sum(a^2))
> dotprod = function(a,b) sum(a*b)
>
> # clip replaces all negative components with 0 and all components > 1 with 1.
> clip = function(a) pmin(pmax(a, c(0,0,0)), c(1,1,1))
>
> # problem 1
> Nb = normalize((Nf+Ng)/2)
> Nb
[1]  0 -1  0
>
> # problem 2
> Iambient = Ia*ka
>
> lb = normalize(L-B)
> Idiffuse_b = kd*Il*dotprod(Nb, lb)
> Idiffuse_b
[1] 0.1909188 0.5656854 0.1414214
>
> vb = normalize(E-B)
> rb = 2*Nb*dotprod(Nb, lb)-lb
> Ispecular_b = ks*Il*dotprod(vb, normalize(rb))^kse
> Ispecular_b
[1] 0.001241068 0.002757930 0.002757930
>
> Itotal_b = clip(Iambient) + clip(Idiffuse_b) + clip(Ispecular_b)
> Itotal_b
[1] 0.4421599 0.5884434 0.3941793
>
> lc = normalize(L-C)
> Idiffuse_c = kd*Il*dotprod(normalize(Nc), lc)
> Idiffuse_c
[1] -0.2465124 -0.7304070 -0.1826017
>
> vc = normalize(E-C)
> rc = 2*normalize(Nc)*dotprod(normalize(Nc), lc)-lc
> rc
```

```
[1] -0.99622642 -0.08679245  0.00000000
> Ispecular_c = ks*Il*dotprod(vc, normalize(rc))^kse
> Ispecular_c
[1] 0.0006910679 0.0015357063 0.0015357063
>
> Itotal_c = clip(Iambient) + clip(Idiffuse_c) + clip(Ispecular_c)
> Itotal_c
[1] 0.25069107 0.02153571 0.25153571
>
> # Since we're using the flat shading model, D is the same as C
> Idiffuse_d = Idiffuse_c
> Idiffuse_d
[1] -0.2465124 -0.7304070 -0.1826017
> Ispecular_d = Ispecular_c
> Ispecular_d
[1] 0.0006910679 0.0015357063 0.0015357063
> Itotal_d = clip(Iambient) + clip(Idiffuse_d) + clip(Ispecular_d)
> Itotal_d
[1] 0.25069107 0.02153571 0.25153571
>
> # problem 3
> # B and C will be the same, since Gouraud shading only affects the normal interpolation.
> percent_c = ((D-B)/(C-B))[1]
> percent_c
[1] 0.6666667
>
> Idiffuse_d = clip(Idiffuse_c)*percent_c + clip(Idiffuse_b)*(1-percent_c)
> Idiffuse_d
[1] 0.06363961 0.18856181 0.04714045
> Ispecular_d = clip(Ispecular_c)*percent_c + clip(Ispecular_b)*(1-percent_c)
> Ispecular_d
[1] 0.0008744014 0.0019431141 0.0019431141
> Itotal_d = clip(Iambient) + Idiffuse_d + Ispecular_d
> Itotal_d
[1] 0.3145140 0.2105049 0.2990836
>
> # problem 4
> # B and C will be the same, since Phong shading only affects the normal interpolation.
> Nd = normalize(normalize(Nb)*(1-percent_c) + normalize(Nc)*percent_c)
>
> ld = normalize(L-D)
> Idiffuse_d = kd*Il*dotprod(Nd, ld)
> Idiffuse_d
[1] -0.1622417 -0.4807163 -0.1201791
>
> vd = normalize(E-D)
> rd = 2*normalize(Nd)*dotprod(normalize(Nd), ld)-ld
> rd
[1] -0.5876880  0.8090876  0.0000000
> Ispecular_d = ks*Il*dotprod(vd, normalize(rd))^kse
> Ispecular_d
[1] 0.0003846106 0.0008546902 0.0008546902
>
> Itotal_d = clip(Iambient) + clip(Idiffuse_d) + clip(Ispecular_d)
> Itotal_d
[1] 0.25038461 0.02085469 0.25085469
```