

- In OpenGL haben Materialien und Lichter **ambiente**, **diffuse**, **spekulare** und **emissive** Farb-Komponenten, die miteinander interagieren; die Summe Σ_Q über die Lichtanteile aller Lichtquellen ergibt (für $j=R,G,B$) die Farbkomponenten:

$$I_j = I_{ej} + I_{Gaj} \cdot \rho_{aj} + \Sigma_Q [I_{Qaj} \cdot \rho_{aj} + I_{Qdj} \cdot \rho_{dj} \cdot (\underline{q} \cdot \underline{n}) + I_{Qsj} \cdot \rho_{sj} \cdot (\underline{r} \cdot \underline{v})^f] , \text{ mit:}$$

I_{ej} **emissive** Farb-Komponente: beleuchtungs-unabhängige Objekt-Farbe konstanter Helligkeit ohne Beleuchtungswirkung (z.B.: Ampel, Leuchtreklame, beleuchtete Fenster etc.)

I_{Gaj} **globale ambiente** Helligkeit, pauschal für die gesamte Szene eingesetzt, ohne Festlegung einer Lichtquelle (Tag/Nacht etc.)

I_{Qaj} **ambiente** Beleuchtungskomponente einer zugeschalteten Lichtquelle, ohne Strahlungsrichtung (Streulicht, z.B. Deckenfluter)

I_{Qdj} **diffuse** Komponente einer L.-Quelle; macht je nach Strahlungsrichtung Flächen heller o. dunkler (Lambert-Gesetz, vgl. Sonne)

I_{Qsj} **spekulare** Quelle-Komponente; stark gerichtetes Licht (Spot)

$\rho_{aj}, \rho_{dj}, \rho_{sj}$: Reflexionskoeffizient; $(\underline{q} \cdot \underline{n}), (\underline{r} \cdot \underline{v})^f$: Lambert-, Phong-Anteile

Schattierung und Beleuchtung

Szene mit emissiven Flächen, ohne Lichtquellen (ca. 1994):



- Licht-Farbwerte $[L_R \ L_G \ L_B]$ sind Anteile additiver Farbmischung;
Bsp.: $L_R=L_G=1, L_B=0 \Rightarrow$ gelb $L_R=L_G=L_B=0.5 \Rightarrow$ schwach weiß
- Material-Farbwerte $[M_R \ M_G \ M_B]$ geben Reflexionsanteile an; bei voller Reflexion (senkrechte Blick- und Lichtrichtung, $\rho=1$) ist schließlich als Farbe sichtbar: $[M_R \cdot L_R; M_G \cdot L_G; M_B \cdot L_B]$;
Bsp.: rote Fläche mit $[M_R \ M_G \ M_B]=[1 \ 0 \ 0]$ erscheint schwarz in blauem Licht mit $[L_R \ L_G \ L_B] = [0 \ 0 \ 1]$. `glutInitDisplayMode(GLUT_RGB);`
- Farbmodus: RGBA (Farbindex komplex, mäßige Ergebnisse)
- Benötigt: Schattierungsmodell `glShadeModel(GL_SMOOTH);`
- Hinzufügen v. Beleuchtung in einem OpenGL-Programm durch:
 1. Festlegung der Flächennormalen an allen Objektecken
 2. Einrichtung, Positionierung, Aktivierung von Lichtquelle/n
 3. Auswahl eines „Beleuchtungsmodells“ (*lighting model*) – d.h. des globalen ambienten Lichts u.d. rechnerischen Augenpunkts
 4. Festlegung der Materialeigenschaften der Objekte in der Szene

- OpenGL erlaubt die Festlegung von mind. 8 Lichtquellen mit vorgegebenen Namen u. voreingestellten (variablen) Lichtfarben `GL_LIGHT0`(weiß)...`GL_LIGHT1`(schwarz)...`GL_LIGHT7`(schwarz).
- Separate Aktivierung für Licht allgemein und für jede Quelle:
`glEnable(GL_LIGHTING);`
`glEnable(GL_LIGHT0); /*GL_LIGHT1, ..., GL_LIGHT7*/`
- Belieb. Lichtquellen-Positionierung über `GLfloat`-Array – z.B.:
`GLfloat LitPos[4]; /*LitPos[0]=..LitPos[1]=...*/`
`glLightfv(GL_LIGHT0, GL_POSITION, LitPos);`
- Quellen: „local“ (auch „positional“: positionsgebunden, Spot) oder „directional“ (überall gleich gerichtet, Sonne) über CF:
`LitPos[4]={0.,0.,1.,1.}; /*Lichtpunkt @(0,0,1)*/`
`LitPos[4]={0.,0.,1.,0.}; /*Licht neg. z-Richtung*/`
(direktionales Licht: weniger Rechenzeit, weniger Realismus)
Angenommene Beleuchtungsrichtung: zum Koord.-Ursprung

Schattierung und Beleuchtung

Bestimmung der Eigenschaften von OpenGL-Lichtquellen mit

```
void glLight{if}[v] (GLenum light, GLenum pname,  
                    TYPE [*]param);
```

mit **light** aus: **GL_LIGHT0** ... **GL_LIGHT7**, **pname** aus:

{...}:eins daraus
[...]:kann fehlen

pname	Default param	Meaning
GL_AMBIENT	(0.,0.,0.,1.)	ambient RGBA light intensity
GL_DIFFUSE	(1.,1.,1.,1.)	diffuse RGBA light intensity
GL_SPECULAR	(1.,1.,1.,1.)	specular RGBA light intensity
GL_POSITION	(0.,0.,1.,0.)	(x, y, z, w) light position
GL_SPOT_DIRECTION	(0.,0.,-1.0)	(x, y, z) spotlight direction
GL_SPOT_EXPONENT	0.0	spotlight exponent
GL_SPOT_CUTOFF	180.0	spotlight cutoff angle
GL_CONSTANT_ATTENUATION	1.0	constant attenuation factor
GL_LINEAR_ATTENUATION	0.0	linear attenuation factor
GL_QUADRATIC_ATTENUATION	0.0	quadratic attenuation factor

Anmerkungen zur Verwendung von Lichtquellen in OpenGL :

- Alpha-Komponente in `RGBA` (def.:1) wird ignoriert, solange kein Blending (Farbmischung und Mischungsfunktion) vereinbart ist.
- Objekte mit übergroßen Flächen können dunkler erscheinen, da Schattierung mit `GL_SMOOTH` zwischen Helligkeitswerten von Eckpunkten interpoliert (Bsp.: Laterne vor Kaserne-Mauer).
- Licht von („local“-)Quellen wird abgeschwächt (engl. *attenuated*) nach: $1/(k_C + k_L d + k_Q d^2)$, mit d für d. Abstand Objekt-Lichtquelle
 - k_C : `GL_CONSTANT_ATTENUATION` (def.: 1.),
 - k_L : `GL_LINEAR_ATTENUATION` (def.: 0.),
 - k_Q : `GL_QUADRATIC_ATTENUATION` (def.: 0.) – rechenintensiv!Emissive u. globale Komponenten werden nicht abgeschwächt.
- Reflexionen-Berechnung über d. „Halfway Vector“: Rechenzeit-Einsparung bei Annahme direktionalen Lichts ($w=0$)
 - ⇒ \underline{q} gleich für alle Objektecken (\underline{q} : Strahl zur Lichtquelle)

Schattierung und Beleuchtung

Auswahl eines „Beleuchtungsmodells“ (*lighting model*): Festlegung der globalen ambienten Lichtstärke u. diverser Berechnungsmodi:

```
void glLightModel{if}[v] (GLenum pname, TYPE [*]param);
```

mit `pname` und `param` gemäß:

{...}:eins daraus
[...] :kann fehlen

<code>pname</code>	Default <code>param</code>	Meaning
<code>GL_LIGHT_MODEL_AMBIENT</code>	<code>(.2,.2,.2, 1.)</code>	ambient RGBA intensity of the entire scene
<code>GL_LIGHT_MODEL_LOCAL_VIEWER</code>	<code>0.</code> or <code>GL_FALSE</code>	how specular reflection angles are computed
<code>GL_LIGHT_MODEL_TWO_SIDE</code>	<code>0.</code> or <code>GL_FALSE</code>	choose between one-sided or two-sided lighting
<code>GL_LIGHT_MODEL_COLOR_CONTROL</code>	<code>GL_SINGLE_COLOR</code>	whether specular color is calculated separately from ambient and diffuse

Zur Rechenzeit-Einsparung: Verzicht auf Beleuchtung abgewandter Objektflächen und Einstellung eines weit entfernten Augenpunkts

Schattierung und Beleuchtung

Bestimmung der Material-Eigenschaften von Objektflächen mit

```
void glMaterial{if}[v] (GLenum face, GLenum pname,  
                      TYPE [*]param);
```

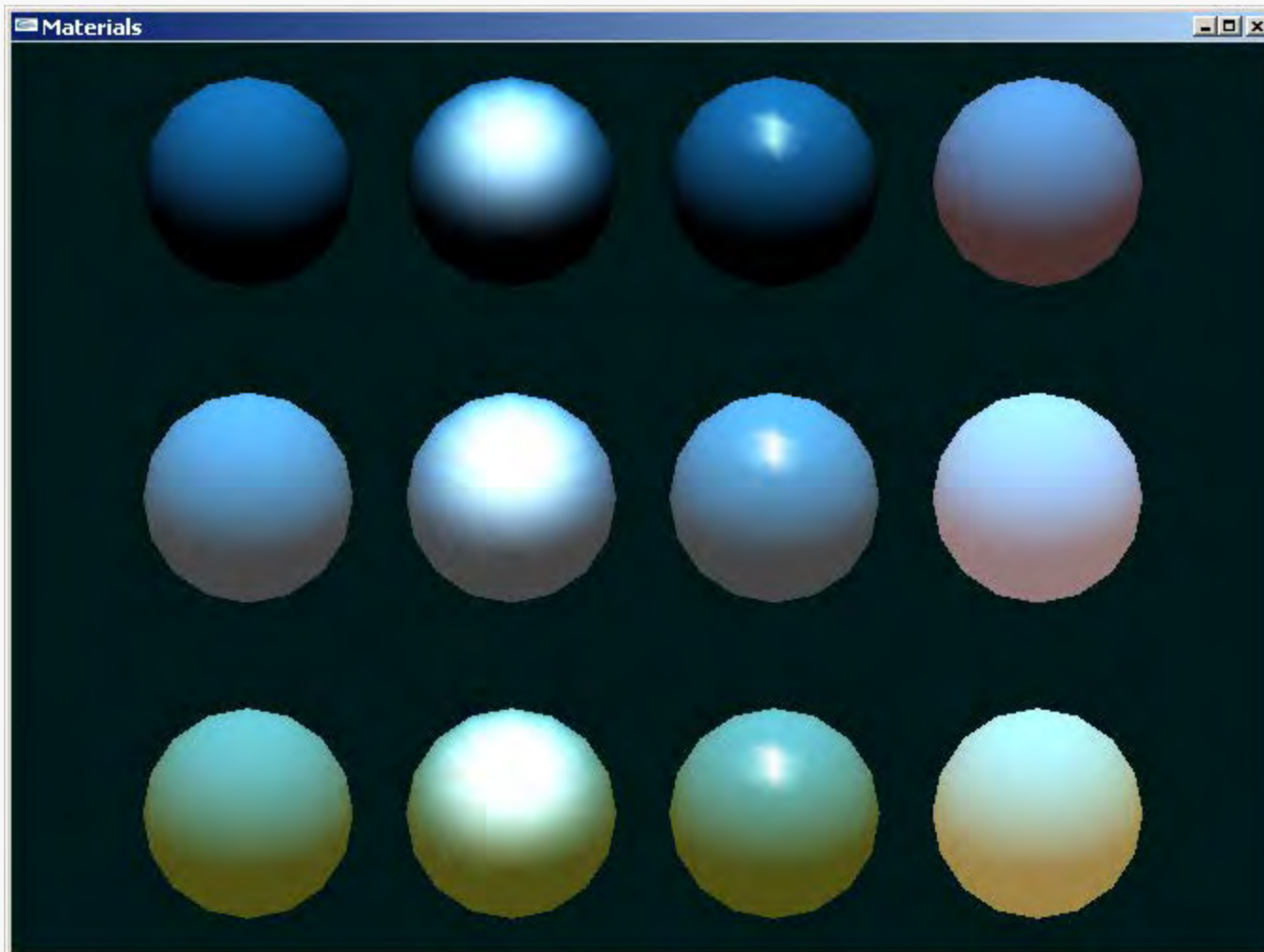
mit **face** aus: `GL_FRONT`, `GL_BACK`, `GL_FRONT_AND_BACK`
und **pname** und **param** gemäß:

{...}:eins daraus
[...]:kann fehlen

Parameter pname	Default param	Meaning
<code>GL_AMBIENT</code>	<code>(.2,.2,.2,1.)</code>	ambient color of material
<code>GL_DIFFUSE</code>	<code>(.8,.8,.8,1.)</code>	diffuse color of material
<code>GL_AMBIENT_AND_DIFFUSE</code>		ambient & diffuse color of material
<code>GL_SPECULAR</code>	<code>(0.,0.,0.,1.)</code>	specular color of material
<code>GL_SHININESS</code>	<code>0.0</code>	specular exponent
<code>GL_EMISSION</code>	<code>(0.,0.,0.,1.)</code>	emissive color of material
<code>GL_COLOR_INDEXES</code>	<code>(0,1,1)</code>	ambient, diffuse, and specular color indices

Schattierung und Beleuchtung

Beispiele von Material-Einstellungen in OpenGL:



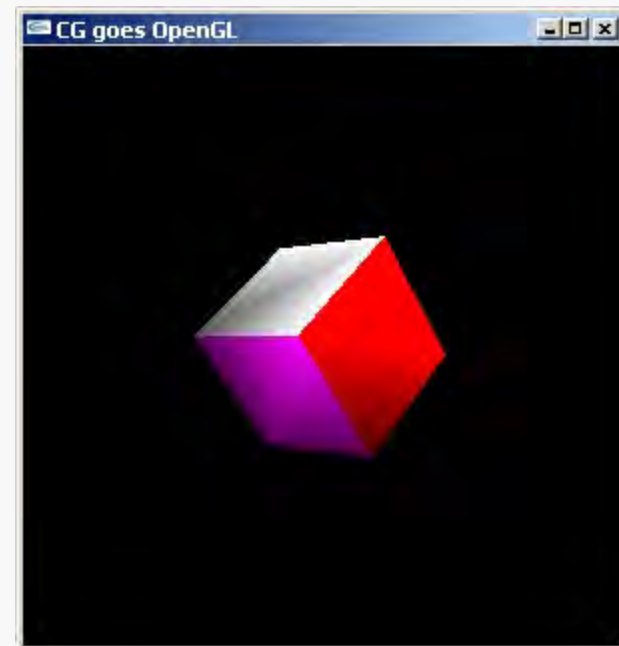
... bei gleichen
Licht-Quelle
und -Modell

Lichtquellen werden wie Objekte positioniert / animiert mittels Transformationen im `glMatrixMode(GL_MODELVIEW)`, d.h.:

- Aufruf von `glLight*(...,GL_POSITION,...)` im Kontext der Objekt- oder Szene-Animation bewegt Lichtquelle mit (vgl. Autoscheinwerfer)
- Unabhängige Licht-Animation (vgl. Sonne, Blaulicht) erfordert getrennte Trfn.; da es genau einen Matrix-Stack gibt, wird dazu
 - die vorausgegangene Objekt-Trf.-Folge zwischengespeichert (`glPushMatrix()`) und ggf. verworfen (`glLoadIdentity()`),
 - eine eigenständige Licht-Animation (mit `glTranslate*()`, `glRotate*()` etc.) aufgebaut,
 - die Quelle (über `glLight*(...,GL_POSITION,...)`) positioniert und schließlich
 - die Transformationen der Objekt-Animation (`glPopMatrix()`) wieder hergestellt.

Übung:

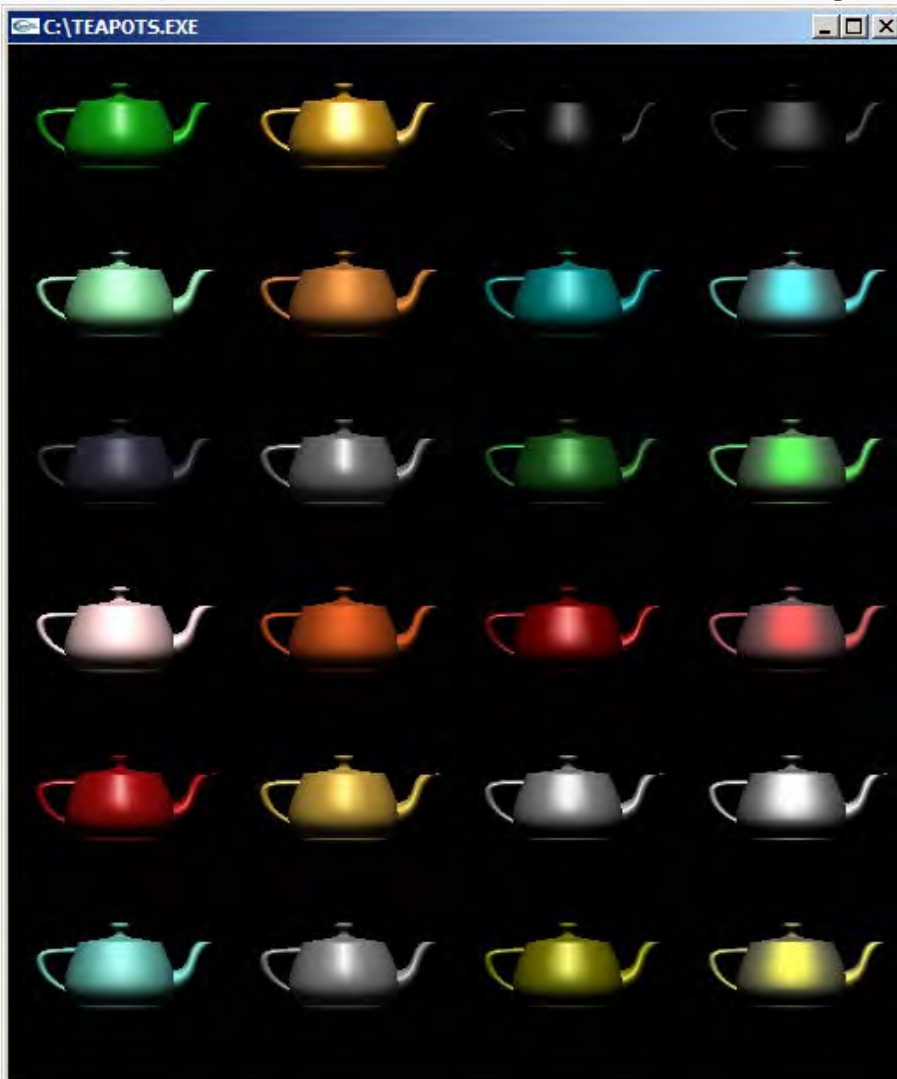
Erweiterung des OpenGL-Programms zur Darstellung eines 3D-Modells um eine animierte direktionale Lichtquelle (s. Übungsblatt).



ObjElabGLaaTx.exe

Schattierung und Beleuchtung

Beispiele von Material-Einstellungen in OpenGL:



... bei gleichen
Licht-Quelle
und -Modell

Zusammenfassung der Lichteffekte:

Farb-Komponente	Strahlung	Reflexion
spekular	gerichtet	gerichtet
diffus	gerichtet	nicht gerichtet
ambient	nicht gerichtet	nicht gerichtet
emissiv	keine	nicht gerichtet

teapots.exe