Graphische Datenverarbeitung II

Visualisierungstechniken I

Prof. Dr. Elke Hergenröther

Visualisierungstechniken

Visualisierung:

Visualisierung bedeutet sichtbar machen, darstellen. Die CG beschränkt sich dabei jedoch nicht auf die Abbildung real existierender Objekte sondern beschäftigt sich auch mit der Darstellung von Information und wissenschaftlich-technischen Daten.

Visualisierung geometrischer Modelle

- Visualisierung ohne Berücksichtigung von Licht
- Visualisierung mit Berücksichtigung von Licht
  - lokale Verfahren
  - globale Verfahren

Visualisierung ohne Berücksichtigung von Licht

- Darstellen eines Primitivs durch setzen eines Farbattributs:
  - rote Fläche,
  - grüne Linie
- Was passiert, wenn jedem Eckpunkt (Vertex) eine andere Farbe zugeordnet wird?

Visualisierung ohne Berücksichtigung von Licht

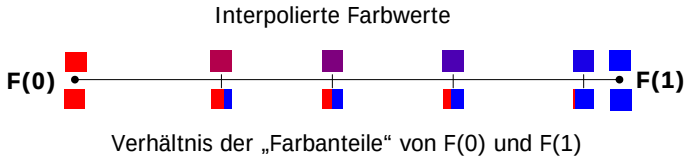
Lineare Interpolation:

Interpolationsverfahren für Linien

Bilineare Interpolation:

Interpolationsverfahren für Flächen

Lineare Farbinterpolation



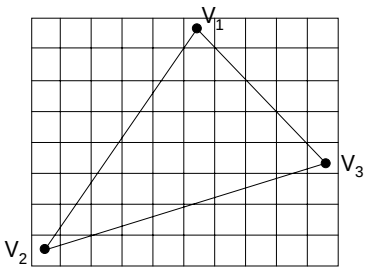
Lineare Farbinterpolation

$$F(t) = (1-t)\cdot F(0) + t \cdot F(1)$$

$$F(t) = (1-t)\cdot \begin{pmatrix} R(0) \\ G(0) \\ B(0) \end{pmatrix} + t \cdot \begin{pmatrix} R(1) \\ G(1) \\ B(1) \end{pmatrix}$$

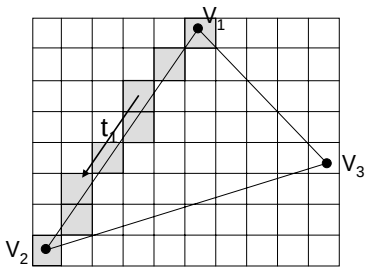
Bilineare Farbinterpolation für Dreiecke

1. Schritt: Rasterisierung des Dreiecks



Bilineare Farbinterpolation für Dreiecke

2. Schritt: Farbwerte der Pixel zwischen den Vertices interpolieren

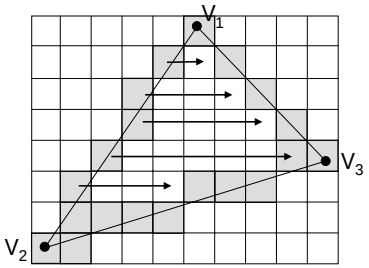


Interpolation zwischen V_1 und V_2:

- Zunächst muss die Strecke zwischen den beiden Vertices normiert werden.
- Der Parameter t_1 steht für die normierte Strecke und geht von 0 bis 1.
- Danach erfolgt eine lineare Farbinterpolation abhängig von t_1.

Bilineare Farbinterpolation für Dreiecke

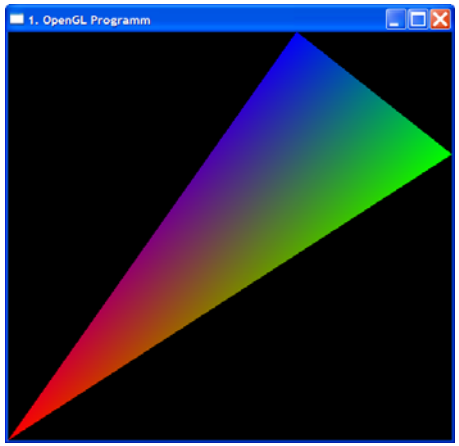
Farbwerte der Pixel zwischen den Vertices wurden linear interpoliert



3. Schritt:

Lineare Interpolation der Pixelwerte entlang der einzelnen Rasterzeilen

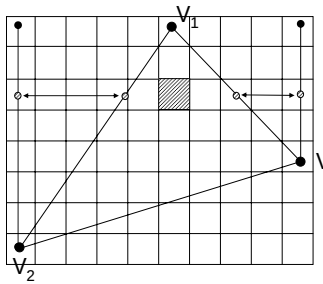
Ergebnis einer bilinearen Farbinterpolation



Farbinterpolation eines Punktes im Dreieck

1. Berechnung erfolgt durch Strahlensatz

$F(P_L)=5,5/8 * V_1 + 2,5/8 * V_2$

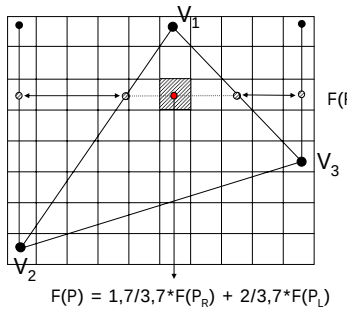


$F(P_R) = 1/2 * V_1 + 1/2 * V_3$

Farbinterpolation eines Punktes im Dreieck

2. Lineare Interpolation

$F(P_L)=5,5/8 * V_1 + 2,5/8 * V_2$



$F(P_R) = 1/2 * V_1 + 1/2 * V_3$

Rendering

Rendern:

- Rendern bedeutet: Visualisieren einer beleuchteten Szene.
- Dabei spielen die Beleuchtungsverhältnisse und die Materialien, die auf die Beleuchtung reagieren eine wichtige Rolle.

Renderer:

- Hard- oder Software, die die Visualisierung der Szene übernimmt.

Faktoren, die das Aussehen eines Objektes bestimmen:

- Position und Orientierung des Betrachters relativ zum beleuchteten Objekt
- Position, Orientierung und Typ der Lichtquelle
- Beschaffenheit der Objektoberfläche
- Verfahren zum Rendern des Objektes

Rendering

Benötigte Information zum Rendern:

Objekt: Position, Orientierung, Materialeigenschaften

Lichtquelle: Lage, Strahlrichtung, Lichtfarbe, Intensität

Betrachter: Position, Blickrichtung (Orientierung)

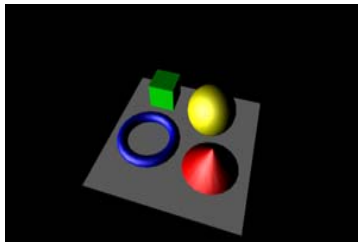
Lichtquellen

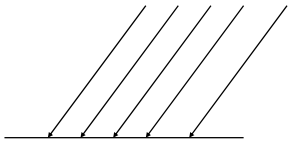
Typischerweise verwendet man 3 verschiedene Lichtquellenarten:

- gerichtete Lichtquelle
- Punktlichtquelle
- Spotlichtquelle

Gerichtete Lichtquelle

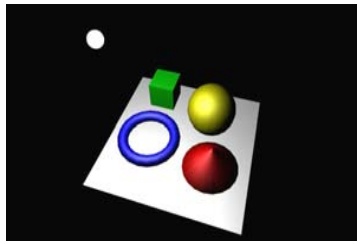
- Entsprechen einer unendlich weit entfernten Lichtquelle
- parallel einfallende Strahlen.
- **Parameter:** Richtung, Farbe

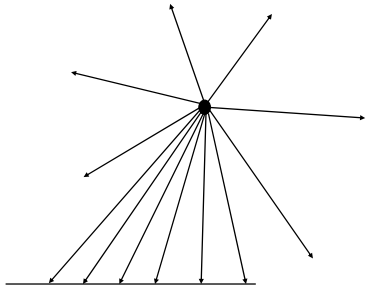




Punktlichtquelle

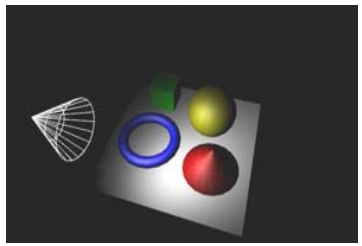
- Ein lokalisierter Punkt strahlt das Licht aus.
- unterschiedliche Winkel auf eine ebene Fläche
- **Parameter:** Position, Farbe





Spotlichtquelle

- Nur um einen bestimmten Winkel um eine angegebene Richtung wird Licht ausgestrahlt.
- je weiter von der Richtung weg desto schwächer wird das Licht ($\cos^n$-Verteilung)
- Parameter: Position, Richtung, Exponent, Farbe



Rendering

Benötigte Information zum Rendern:

Objekt: Position, Orientierung, Materialeigenschaften

Lichtquelle: Lage, Strahlrichtung, Lichtfarbe, Intensität

Betrachter: Position, Blickrichtung (Orientierung)

Materialeigenschaften des Objektes

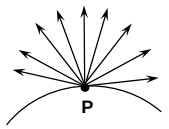
Abhängig von den Materialeigenschaften des Objektes werden die Reflexionseigenschaften bestimmt werden:

- Diffuse Reflexion
- Spiegelnde Reflexion

Ab hier weiter .....

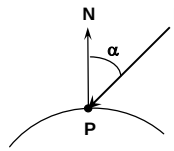
Diffuse Reflexion

- Die Reflexion des Lichtes ist in alle Richtungen gleich.
- Position des Betrachters hat daher keinen Einfluss auf die Berechnung der Reflexion.



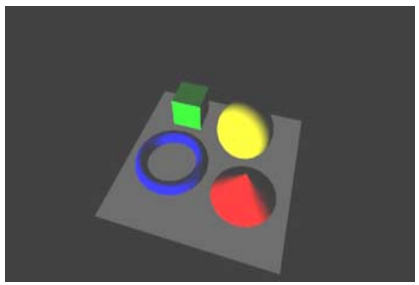
Lichtabstrahlung

- Die Reflexion ist nur abhängig von der Normalen und der Lichtrichtung.
- je größer α, desto dunkler die Facette
- Beleuchtungsberechnung nur abhängig von der Normalen und dem einfallenden Licht



Berechnung der Reflexion

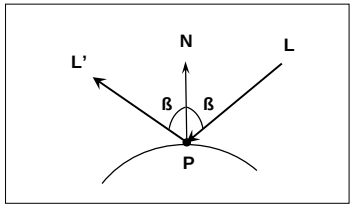
Diffuse Reflexion



- Sehr matte Oberflächen sind perfekt diffus reflektierend.
- Diese Oberflächen haben unabhängig von der Position des Betrachters die gleiche Farbe und Helligkeit.
- Farbe der diffusen Reflexion ist Materialabhängig.

Spiegelnde Reflexion

- Das von der Oberfläche reflektierte Licht verlässt diese nur unter dem Winkel β (Einfallswinkel = Ausfallswinkel)
- Damit ist das reflektierte Licht nur von einer bestimmten Betrachterposition aus sichtbar.
- Für alle anderen Positionen erscheint die Oberfläche dunkel.

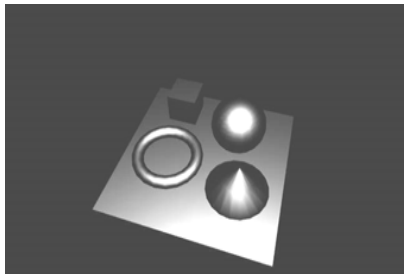


L = einfallendes Licht

L' = reflektiertes Licht

β = Einfalls- und Ausfallswinkel

Spiegelnde Reflexion



Im Gegensatz zum diffus reflektierenden Licht hat spiegelnd reflektierendes Licht nicht die Farbe der Oberfläche sondern die des Lichtes.

Beispiel: Ein grünes Objekt wird mit weißen Licht bestrahlt.

- Das diffus reflektierende Licht ist grün.
- Das spiegelnd reflektierendes Licht ist weiß.

Licht aus der Umgebung

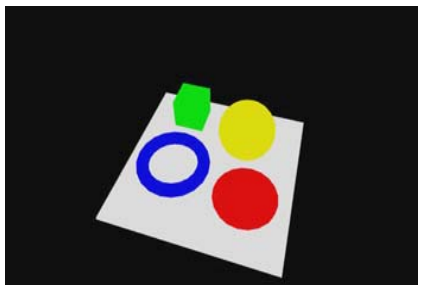
- Oberflächen, die parallel zum einfallenden Licht liegen, haben keine diffuse Reflexion ...
- ... und erst recht keine spiegelnde Reflexion.
- D.h. sie müssten schwarz dargestellt werden.
- Realistisch ist aber, dass diese Flächen vom abgestrahlten Licht der Umgebung beleuchtet werden also auch sichtbar sind.

Simulation dieses Effektes durch Anwendung eines Umgebungslichts:

Der Ambienten Materialbeleuchtung

Ambiente Materialbeleuchtung

- Die Ambiente Beleuchtung soll das Umgebungslicht simulieren.
- Sie ist nur von der Materialfarbe und der eingestellten Lichtintensität abhängig.



Verfahren zum Rendering

Benötigte Information zum Rendern:

Objekt: Position, Orientierung, Materialeigenschaften

Lichtquelle: Lage, Strahlrichtung, Lichtfarbe, Intensität

Betrachter: Position, Blickrichtung (Orientierung)

Verfahren zum Rendering (Beleuchtungsberechnung):

- Umfangreiche Daten müssen zu Formeln verknüpft werden.
- **Resultat:** Aufwendige Berechnungen.
- **Gesucht:** Trade off zwischen Berechnungszeit und relativ akzeptablem visuellen Ergebnis.

Verfahren zum Rendering

Lokale Verfahren:

- Berücksichtigt bei der Berechnung des von der Objektoberfläche reflektierten Lichts nur das von der definierten Lichtquelle einfallende Licht (primäre Lichtquellen).
- Die Beleuchtung der Objekte wird nicht beeinflusst durch:
  - Spiegelung,
  - Schattenwurf und
  - Lichtreflexion anderer Objekte.

  } sekundäre Lichtquellen

Verfahren zum Rendering

Globale Verfahren:

- Objekte der Szene beeinflussen sich gegenseitig.
- In die Beleuchtungsberechnung gehen neben den primären Lichtquellen auch die sekundäre mit ein.
- D.h. in der Beleuchtungsberechnung werden berücksichtig:
  - Spiegelung,
  - Schattenwurf und
  - Lichtreflexion anderer Objekte.

Verfahren zum Rendering

Lokale Verfahren

- Wenig rechenaufwendig - schnelle Berechnung
- Beispiele: Flat Shading
  Gouraud Shading
  Phong Shading

Globale Verfahren

- Sehr rechenaufwendig
- Beispiele: Rekursives Ray Tracing
  Radiosity

Was heißt eigentlich Shading?

Einfärben einer Facettenoberfläche (polygonalen Oberfläche) entsprechend den Material-, Beleuchtungs- und Betrachtungsverhältnissen.

Shading Verfahren

Mit den folgenden Shading Verfahren können prinzipiell diffuse **und** spiegelnde Lichtreflektionen berechnet werden.

Flat und Gouraud Shading eignet sich nicht sehr gut zur Berechnung einer spiegelnden Lichtreflektion.

Die Stärke von Phong Shading hingegen ist die Berechnung von Glanzlichtern, also spiegelnden Lichtreflektionen.

Der Unterschied der Verfahren liegt im wesentlichen in der Anzahl der Normalen, die pro Facette zur Beleuchtungsberechnung verwendet werden:

zunehmende Anzahl von Normalen pro Facette ↓

- Flat Shading
- Gouraud Shading
- Phong Shading

Flat Shading

Verfahren:

- Jede Facette (Polygon) erhält eine einheitliche Helligkeit.
- Für jede Facette gibt es nur eine Normale.
- Aufgrund dieser Normalen wird die Helligkeit der gesamten Facette berechnet.
- D.h. Jede Facette besitzt **nur eine Farbe**.

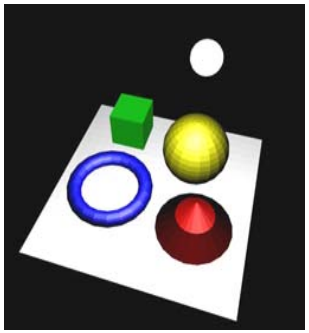
Nachteil:

- Da die meisten Objektoberflächen gekrümmt sind, ist die Qualität der Ergebnisse meist schlecht.

Vorteil:

- sehr einfache und sehr schnelle Beleuchtungsberechnung

Flat Shading



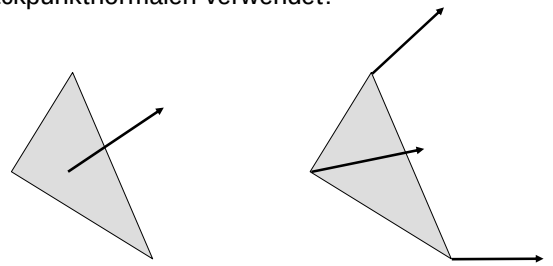
Flat Shading

Voraussetzung für brauchbare Ergebnisse:

- Lichtquelle sehr weit entfernt vom Betrachter
- Betrachter sehr weit von der Lichtquelle entfernt
- Oberfläche ist eben

Gouraud Shading

Zur Beleuchtungsberechnung werden keine Flächennormalen sondern Eckpunktnormalen verwendet:



Flächennormale

Eckpunktnormalen

(= Eckennormalen)

Gouraud Shading

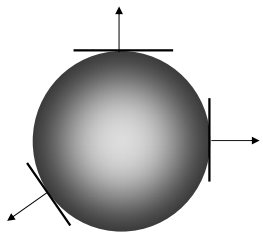
- Beleuchtungsberechnung wird aufgrund der Eckpunktnormalen durchgeführt.
- D.h. durch die Beleuchtungsberechnung werden pro Dreieck drei unterschiedliche Farben (entspricht drei unterschiedlichen Helligkeiten einer Farbe) berechnet.
- Die Farben der Pixel innerhalb des Dreiecks werden durch die bilineare Interpolation berechnet.
- Dadurch entsteht eine **optische Glättung** der in einzelne Facetten zerlegten Oberfläche – **ohne** Änderungen an der Geometrie vornehmen zu müssen.

Gouraud Shading

Definition der Eckpunktnormalen:

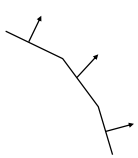
Eckpunktnormalen werden an der Tangentialebene ausgerichtet.

Beispiel Kugel:



Dadurch werden Kanten weniger sichtbar.

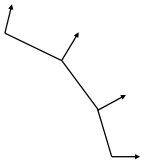
Gouraud Shading



Ergebnis ist abhängig von den berechneten Normalen:

Links: Flächennormalen

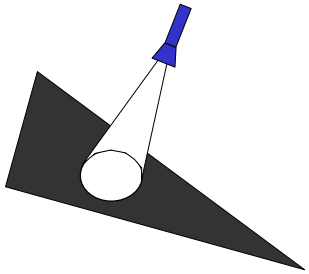
Rechts: Eckpunktnormalen



Gouraud Shading

Grenzen des Verfahrens:

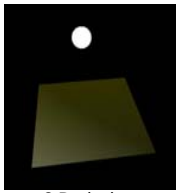
- Stark gebündelter Strahl bestrahlt nur die Mitte eines Dreiecks.
- Da die Lichtberechnung nur an den Eckpunkten berechnet wird, wird für das Dreieck keine Reflektion berechnet.



Gouraud Shading

- da nur an den Eckpunkten die Beleuchtung wirklich berechnet wird, hängt die Beleuchtungsgüte von der Anzahl der verwendeten Polygone ab
- dies gilt auch für völlig ebene Flächen!

Gouraud Shading



2 Dreiecke

8 Dreiecke

18 Dreiecke

50 Dreiecke

200 Dreiecke

3200 Dreiecke

Phong Shading

Verfahren:

- Die Beleuchtungsberechnung wird für jedes Pixel einzeln durchgeführt (also Betrachterabhängig!).
- Für jedes Pixel muss also eine Normale berechnet werden.
- Die Normalen der Pixel werden auf Basis der Eckpunktnormalen ermittelt (bilineare Interpolation).
- Oder durch eine Art Textur (Normal Map).
- Dadurch können Glanzlichter ohne zusätzliche Unterteilung (Tessilierung) des Gitters berechnet werden.

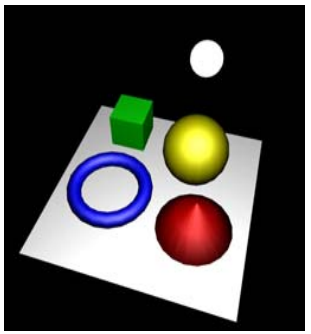
Vorteil:

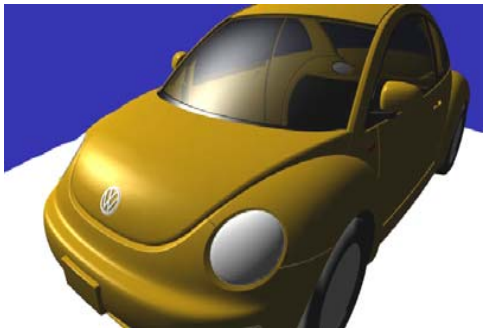
- Glanzlichter & gutes visuelles Ergebnis

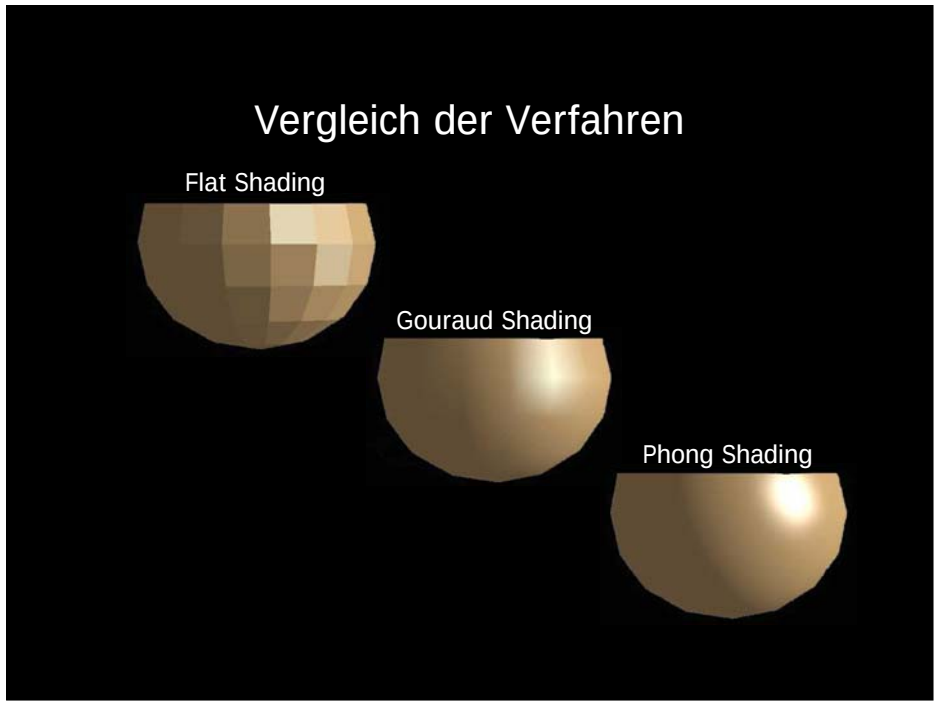
Nachteil:

- Normale für jeden Pixel interpolieren kostet viel Rechenzeit

Phong Shading







Kurze Einführung: Lichtberechnung in OpenGL

- Ohne Lichtberechnung (Programmaufbau)
- Mit Umgebungslicht
- Mit diffuser Beleuchtungsberechnung
- Flat und Gouraud Shading

Nicht Klausurrelevant

Ohne Lichtberechnung:

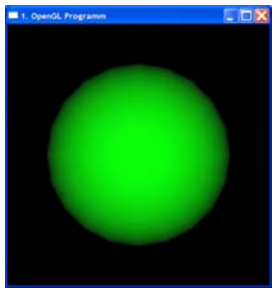
```
void display()
{
   glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
   glColor4f(1.f,.0f,.0f,1.0f);
   glutSolidSphere( 0.7, 20, 20 );
   glFlush();
}

int main(int argc, char **argv)
{
   glutInit(&argc, argv);
   glutInitDisplayMode( GLUT_DEPTH |GLUT_SINGLE |GLUT_RGB ); //
   glutInitWindowSize(500, 500);
   glutCreateWindow("1. OpenGL Programm");
   // Initialisierung der Lichtquellen und setzen des Sphere-Materials
   //Init();
   glutDisplayFunc(display);
   glutMainLoop();
   return 0;
}
```

Mit Umgebungslicht (default: ausgeschaltet)

```
void Init()
{
   GLfloat mat_ambient[] = { 1., 1., 0., 1.};
   GLfloat lmodel_ambient[] = {1., 1., 1., 1.}; // Ambientes Licht
   glClearColor( 0., 0., 0., 0. );
   glMaterialfv( GL_FRONT, GL_AMBIENT, mat_ambient );
   glLightModelfv( GL_LIGHT_MODEL_AMBIENT, lmodel_ambient );
   glEnable( GL_LIGHTING );
   glEnable( GL_DEPTH_TEST );
}
void display()
{
   glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
   glColor4f(1.f,.0f,.0f,1.0f); // wird wirkungslos!
   glutSolidSphere( 0.7, 20, 20 ); //Normalen werden hier automatisch berechnet
   glFlush();
}
int main(int argc, char **argv)
{
   ...
   Init();
   ...
}
```

Mit diffuser Beleuchtungsberechnung

```
void Init()
{
   GLfloat mat_diffus[] = { 0., 1., 0., 1.}; // grüne Kugel
   GLfloat light_position[] = { 0., 0., -7., 0. };
   GLfloat white_light[] = { 1., 1., 1., 1.};
   glClearColor( 0., 0., 0., 0. );
   glMaterialfv( GL_FRONT, GL_DIFFUSE, mat_diffus );
   glLightfv( GL_LIGHT0, GL_POSITION, light_position );
   glLightfv( GL_LIGHT0, GL_DIFFUSE, white_light ); // default
   glEnable( GL_LIGHTING );
   glEnable( GL_LIGHT0 );
   glEnable( GL_DEPTH_TEST );
}
void display()
{
...
}
int main(int argc, char **argv)
{
   ...
   Init();
   ...
}
```

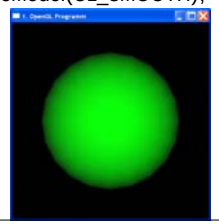


Mit diffuser Beleuchtungsberechnung

```
void Init()
{
   GLfloat mat_diffus[] = { 0., 1., 0., 1.};
   GLfloat light_position[] = { 0., 0., -7., 0. };
   GLfloat white_light[] = { 1., 1., 1., 1.};
   glClearColor( 0., 0., 0., 0. );
   glShadeModel( GL_FLAT ); // Default: GL_SMOOTH
   glMaterialfv( GL_FRONT, GL_DIFFUSE, mat_diffus );
   glLightfv( GL_LIGHT0, GL_POSITION, light_position );
   glLightfv( GL_LIGHT0, GL_DIFFUSE, white_light ); // default
   glEnable( GL_LIGHTING );
   glEnable( GL_LIGHT0 );
   glEnable( GL_DEPTH_TEST );
}
void display()
{
...
}
int main(int argc, char **argv)
{
   ...
   Init();
   ...
}
```

Gouraud Shading: glShadeModel(GL_SMOOTH);

Flat Shading: glShadeModel(GL_Flat);



Zusammenfassung: Beleuchtungsberechnung

- OpenGL unterstützt nur Flat und Gouraud Shading.
- Nochmal die Verfahren im Vergleich:
  Flat, Gouraud und Phong Shading
- Ausblick

Vergleich der Verfahren

Aus: Watt A., Policarpo F.: The Computer Image, Addison-Wesley 1998

Nicht berechnet werden kann mit den hier vorgestellten Shading Verfahen:

- Schattenwurf,
- Transparente Oberflächen,
- sowie spiegelnde Oberflächen

Ausblick: Ray Tracing