



Politechnika  
Wrocławska

# Grafika komputerowa i komunikacja człowiek-komputer

Laboratorium nr 5  
Oświetlanie scen

Szymon Datko  
szymon.datko@pwr.edu.pl

Wydział Informatyki i Telekomunikacji,  
Politechnika Wrocławska

semestr zimowy 2021/2022



# Cel ćwiczenia

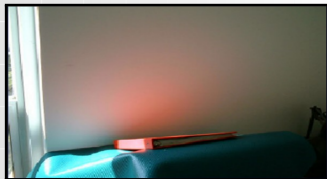
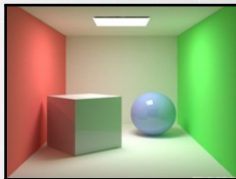
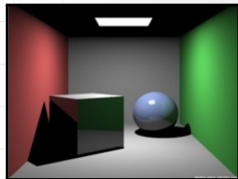
1. Zapoznać się i zrozumieć zawłości zagadnienia oświetlania scen.
2. Zgłębić zasadę działania przykładowego modelu oświetlenia – Phong.
3. Nauczyć się jak programowo obsługiwać źródła światła w OpenGL.
4. Poznać sposób definiowania wektorów normalnych dla własnych modeli.

# Szereg złożonych zjawisk

- ▶ Temat oświetlenia obejmuje bardzo wiele zagadnień:
  - odwzorowanie kolorów obiektów,
  - uwzględnienie parametrów materiałowych,
  - półprzeźroczystość i efekty załamania,
  - cieniowanie i przesłanianie obiektów,
  - efekty cząsteczkowe i wolumetryczne,
  - paralaksa, rozmycia, rozproszenia, itp.
- ▶ Te zjawiska bardzo trudno realizować w czasie rzeczywistym.
- ▶ Dlatego stosuje się szereg odwzorowań uproszczonych...

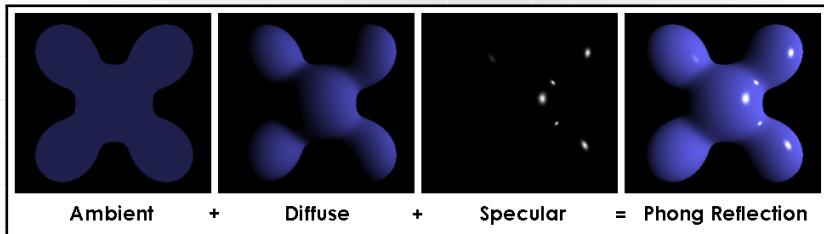
# Rodzaje oświetlenia

- ▶ Oświetlenie lokalne:
  - uwzględnia tylko bezpośredni wpływ źródła światła,
  - może uwzględniać elementy oświetlenia globalnego,
  - realizowane przez uproszczone modele oświetlenia.
- ▶ Oświetlenie globalne:
  - uwzględnia interakcje pomiędzy obiektami,
  - rzucanie cienia, załamania, wielokrotne odbicia, itp.,
  - realizowane na przykład metodami śledzenia promieni, śledzenia ścieżek, mapowania fotonowego i energetycznymi.



# Model Phong

- ▶ Uwzględnia 3 rodzaje oświetlenia, aby odwzorować efekty:
  - światło **kierunkowe** – refleksy odbite zgodnie z prawem Snella,
  - światło **rozproszone** – wpływ bezpośredniego oświetlenia,
  - światło **otoczenia** – jednorodnie oświetlające cały obiekt,
- ▶ dobrze oddaje wygląd obiektów z tworzyw sztucznych.



# Model Phongi – formuła

- ▶ Każda składowa koloru  $C$  w modelu RGB jest obliczana jako

$$C = k_a \cdot I_a + \frac{k_d \cdot I_d \cdot (\vec{N} \circ \vec{L}) + k_s \cdot I_s \cdot (\vec{R} \circ \vec{V})^n}{a + b \cdot d + c \cdot d^2}.$$

- ▶ Parametry modelu i oznaczenia w powyższym równaniu:

- |   |  |
|---|--|
| – $I_s$ - kolor światła (składowa kierunkowa),  | – $k_s$ - kolor materiału (składowa kierunkowa),   |
| – $I_d$ - kolor światła (składowa rozproszona), | – $k_d$ - kolor materiału (składowa rozproszona),  |
| – $I_a$ - kolor światła (składowa otoczenia),   | – $k_a$ - kolor materiału (składowa otoczenia),    |
| – $a, b, c$ - współczynniki strat natężenia.    | – $n$ - współczynnik połysku materiału,            |
| – $\vec{N}$ - wektor normalny w punkcie,        | – $p$ - położenie analizowanego punktu.            |
| – $\vec{V}$ - kierunek do obserwatora,          | – $\vec{L}$ - kierunek padania światła na punkt,   |
| – $d$ - odległość punktu od obserwatora,        | – $\vec{R}$ - kierunek światła odbitego w punkcie, |

( $\vec{V}$  i  $d$  wyznaczamy na podstawie położenia obserwatora,  $\vec{L}$  i  $\vec{R}$  na podstawie położenia źródła).

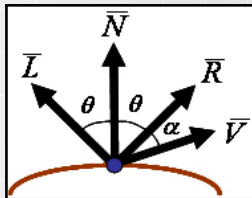
- ▶ Powyższe równania implementuje się w shaderze fragmentów.
- ▶ Faktycznych parametrów w modelu jest 14.

# Model Phongi – uwagi

- ▶ Aby uzyskać najlepsze efekty:
  - wartość składowej kierunkowej powinna być największa,
  - poziom składowej rozproszonej odrobinę mniejszy,
  - składowa otoczenia powinna być bardzo mała.
- ▶ Część obliczeń można przeprowadzić w shaderze wierzchołków,
  - typowo realizuje się je w shaderze fragmentów,
  - da nawet przenieść się wszystkie obliczenia,
  - uzyskuje się wtedy model Gourauda.
- ▶ Niektóre transformacje obiektu wpływają na wektory normalne.
- ▶ Model nie uwzględnia w ogóle cieniowania, pozwala jedynie określić kolor.
- ▶ Trudność może sprawić dobranie parametrów modelu tak, aby odwzorować dobrze powierzchnie metaliczne i anizotropowe.

## Dygresja na temat wektorów normalnych

- ▶ Wskazują kierunek prostopadły do powierzchni w danym punkcie.
- ▶ Stosowane są do obliczenia kierunku światła odbitego  $\vec{R}$ .
- ▶ Poprawność obliczeń wymaga, aby  $\vec{N}$  miał długość 1.
- ▶ W Legacy OpenGL wektor przypisuje się do wierzchołka,
  - analogicznie jak do tej pory określaliśmy kolor wierzchołka,
  - stosuje się funkcję `glNormal()`, przyjmującą 3 współrzędne.





## Nowości w przykładowym programie (1/3)

- Dodano szereg zmiennych pomocniczych.

```
1| mat_ambient = [1.0, 1.0, 1.0, 1.0]
2| mat_diffuse = [1.0, 1.0, 1.0, 1.0]
3| mat_specular = [1.0, 1.0, 1.0, 1.0]
4| mat_shininess = 20.0
5|
6| light_ambient = [0.1, 0.1, 0.0, 1.0]
7| light_diffuse = [0.8, 0.8, 0.0, 1.0]
8| light_specular = [1.0, 1.0, 1.0, 1.0]
9| light_position = [0.0, 0.0, 10.0, 1.0]
10|
11| att_constant = 1.0
12| att_linear = 0.05
13| att_quadratic = 0.001
```

- Odpowiadają one większości parametrów z modelu Phong'a.
- Pierwsze trzy (linie 1-3) opisują składowe koloru materiału.
- Kolejna (linia 4) określa stopień połyskliwości materiału.
- Następne trzy (linie 6-8) mówią o kolorze źródła światła.
- Dalej (linia 9) jest zmienna, opisująca położenie źródła światła.
- Ostatnie 3 zmienne (linie 11-13) określają składowe funkcji strat natężenia.

## Nowości w przykładowym programie (2/3)

- Dodano funkcje związane z uaktywnieniem modelu oświetlenia.

```
1| def startup():
2|     ...
3|
4|     glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient)
5|     glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse)
6|     glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular)
7|     glMaterialf(GL_FRONT, GL_SHININESS, mat_shininess)
8|
9|     glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient)
10|    glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse)
11|    glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular)
12|    glLightfv(GL_LIGHT0, GL_POSITION, light_position)
13|
14|    glLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION, att_constant)
15|    glLightf(GL_LIGHT0, GL_LINEAR_ATTENUATION, att_linear)
16|    glLightf(GL_LIGHT0, GL_QUADRATIC_ATTENUATION, att_quadratic)
17|
18|    glShadeModel(GL_SMOOTH)
19|    glEnable(GL_LIGHTING)
20|    glEnable(GL_LIGHT0)
```

- Warianty wektorowe oczekują 4-elementowych tablic jako argumentów.
- Identyfikator `GL_LIGHT0` wskazuje konkretne źródło światła.

## Nowości w przykładowym programie (3/3)

- ▶ W funkcji `render()` wykonano rysowanie sfery.

```
1| def render(time):
2|     global theta
3|
4|     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
5|     glLoadIdentity()
6|
7|     gluLookAt(viewer[0], viewer[1], viewer[2],
8|               0.0, 0.0, 0.0, 0.0, 1.0, 0.0)
9|
10|    if left_mouse_button_pressed:
11|        theta += delta_x * pix2angle
12|
13|    glRotatef(theta, 0.0, 1.0, 0.0)
14|
15|    quadric = gluNewQuadric()
16|    gluQuadricDrawStyle(quadric, GLU_FILL)
17|    gluSphere(quadric, 3.0, 10, 10)
18|    gluDeleteQuadric(quadric)
19|
20|    glFlush()
```

- Warto zwrócić uwagę, iż z uwagi na określenie materiału, znaczenie tracą teraz wszystkie wywołania `glColor()` – ale nie trzeba ich usuwać z kodu.

# Poruszanie źródłem światła dookoła obiektu

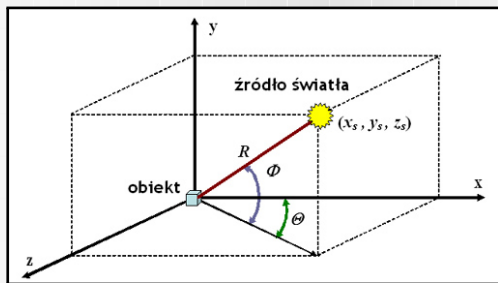
- Współrzędne źródła można określić za pomocą następujących równań,

$$x_s(R, \theta, \phi) = R \cdot \cos(\theta) \cdot \cos(\phi),$$

$$y_s(R, \theta, \phi) = R \cdot \sin(\phi),$$

$$z_s(R, \theta, \phi) = R \cdot \sin(\theta) \cdot \cos(\phi).$$

- Zakresy wartości kątów  $\theta$  i  $\phi$  to przedziały  $0 \leq \theta \leq 2\pi$  oraz  $0 \leq \phi \leq 2\pi$ .
- Parametr  $\theta$  to tak zwany kąt azymutu, zaś  $\phi$  to tak zwany kąt elewacji.



# Wektory normalne modelu z poprzednich zajęć (1/2)

- ▶ Współrzędne wierzchołków jajka można było obliczyć z układu równań,

$$x(u, v) = (-90 \cdot u^5 + 225 \cdot u^4 - 270 \cdot u^3 + 180 \cdot u^2 - 45 \cdot u) \cdot \cos(\pi \cdot v),$$

$$y(u, v) = 160 \cdot u^4 - 320 \cdot u^3 + 160 \cdot u^2 - 5,$$

$$z(u, v) = (-90 \cdot u^5 + 225 \cdot u^4 - 270 \cdot u^3 + 180 \cdot u^2 - 45 \cdot u) \cdot \sin(\pi \cdot v),$$

gdzie dziedziny  $u$  i  $v$  to przedziały  $0 \leq u \leq 1$  oraz  $0 \leq v \leq 1$ .

- ▶ Wektor normalny do powierzchni modelu w punkcie odpowiadającym konkretnym wartościom  $u$  i  $v$  można wyznaczyć przy pomocy wyrażenia,

$$\begin{aligned}\vec{N}(u, v) &= \left[ \begin{array}{c|c} y_u & z_u \\ \hline y_v & z_v \end{array}, \begin{array}{c|c} z_u & x_u \\ \hline z_v & x_v \end{array}, \begin{array}{c|c} x_u & y_u \\ \hline x_v & y_v \end{array} \right] \\ &= [y_u \cdot z_v - z_u \cdot y_v, \quad z_u \cdot x_v - x_u \cdot z_v, \quad x_u \cdot y_v - y_u \cdot x_v],\end{aligned}$$

gdzie elementy  $x_u, x_v, y_u, y_v, z_u$  i  $z_v$  to pochodne cząstkowe...

## Wektory normalne modelu z poprzednich zajęć (2/2)

$$x_u = \frac{\partial x(u, v)}{\partial u} = (-450 \cdot u^4 + 900 \cdot u^3 - 810 \cdot u^2 + 360 \cdot u - 45) \cdot \cos(\pi \cdot v),$$

$$x_v = \frac{\partial x(u, v)}{\partial v} = \pi \cdot (90 \cdot u^5 - 225 \cdot u^4 + 270 \cdot u^3 - 180 \cdot u^2 + 45 \cdot u) \cdot \sin(\pi \cdot v),$$

$$y_u = \frac{\partial y(u, v)}{\partial u} = 640 \cdot u^3 - 960 \cdot u^2 + 320 \cdot u,$$

$$y_v = \frac{\partial y(u, v)}{\partial v} = 0,$$

$$z_u = \frac{\partial z(u, v)}{\partial u} = (-450 \cdot u^4 + 900 \cdot u^3 - 810 \cdot u^2 + 360 \cdot u - 45) \cdot \sin(\pi \cdot v),$$

$$z_v = \frac{\partial z(u, v)}{\partial v} = -\pi \cdot (90 \cdot u^5 - 225 \cdot u^4 + 270 \cdot u^3 - 180 \cdot u^2 + 45 \cdot u) \cdot \cos(\pi \cdot v),$$

► Uwaga, obliczoną wartość  $\vec{N}(u, v)$  na koniec należy jeszcze znormalizować!

Koniec wprowadzenia.

# Zadania do wykonania...

# Zadania do wykonania (1)

Na ocenę **3.0** należy wprowadzić drugie źródło światła.

Wskazówki:

- przestudiować w jaki sposób dodano pierwsze źródło światła,
- nie trzeba definiować ponownie parametrów materiałowych,
- nowemu źródłu nadać inny kolor i położenie przestrzenne,
- drugie źródło będzie identyfikowane przez `GL_LIGHT1`.



## Zadania do wykonania (2)

(po zrealizowaniu zadania poprzedniego)

Na ocenę **3.5** należy umożliwić dynamiczną zmianę składowych koloru światła.

Wskazówki:

- celem jest zaobserwowanie jaki jest wpływ poszczególnych składowych,
- można ograniczyć się wyłącznie do jednego źródła światła,
- zmiana wartości powinna odbywać się za pośrednictwem klawiatury,
- na przykład:
  - ▶ jednym klawiszem wybrać aktualnie zmienianą składową,
  - ▶ dwoma innymi klawiszami zmieniać tę wartość o 0.1 w górę lub w dół,
  - ▶ minimalna wartość składowej koloru to 0.0, a maksymalna 1.0,
  - ▶ pomocniczo – bieżące wartości można wypisywać w konsoli.

## Zadania do wykonania (3)

(po zrealizowaniu zadania poprzedniego)

Na ocenę **4.0** należy dodać poruszanie źródłami światła i ich wizualizację.

Wskazówki:

- wizualizację można wykonać za pomocą sfery zbudowanej z linii,

```
1| quadric = gluNewQuadric()
2| gluQuadricDrawStyle(quadric, GLU_LINE)
3| gluSphere(quadric, 0.5, 6, 5)
4| gluDeleteQuadric(quadric)
```

- użyć wartości  $x_s$ ,  $y_s$  i  $z_s$  jako argumentów funkcji `glTranslate()`,

- wartości `theta` i `phi` pobierać z ruchu myszką, jak w ramach Lab4,

- pamiętać o odwróceniu transformacji po zwizualizowaniu położenia,

- pozycję danego źródła światła do obliczeń koloru ustala wywołanie `glLightfv(GL_LIGHT0, GL_POSITION, light_position)`,

- ▶ należy dodać odpowiednie wywołanie w ramach funkcji `render()`,
- ▶ poprawnie uwzględnić położenie względem bieżących transformacji!

## Zadania do wykonania (4)

(po zrealizowaniu zadania poprzedniego)

Na ocenę **4.5** należy dodać wektory normalne do modelu jajka.

Wskazówki:

- zadanie wymaga zrealizowania wcześniej co najmniej zadania (3) z Lab3,
- wyznaczyć wartości wektorów normalnych dla każdego wierzchołka jajka,
- pamiętać o znormalizowaniu długości wyznaczonych wektorów,
- skojarzyć konkretne wektory normalne z wierzchołkami modelu, używając funkcji `glNormal()` tuż przed `glVertex()`,
- na połowie modelu oświetlenie powinno zachowywać się poprawnie.

# Zadania do wykonania (5)

(po zrealizowaniu zadania poprzedniego)

Na ocenę **5.0** należy wyświetlić wektory normalne i je poprawić.

Wskazówki:

- do wizualizacji użyć prymitywu `GL_LINES`,
  - ▶ rysować od wierzchołka do sumy wierzchołka i wektora normalnego,
- wizualizacja powinna być móc włączana i ukrywana na żądanie,
- wektory normalne na drugiej połówce modelu należy odwrócić,
  - ▶ każdą składową przemnożyć przez wartość  $-1$ ,
- zwrócić także uwagę na ułożenie wektorów na biegunach bryły.