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 2022/2023

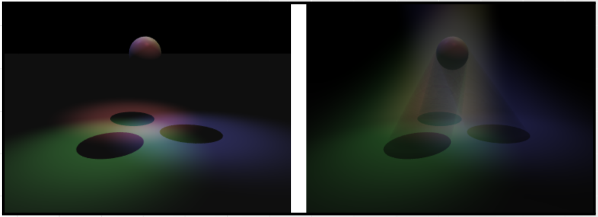
Politechnika Wrocławska

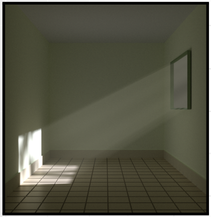
Cel ćwiczenia

1. Zapoznać się i zrozumieć zawiłości zagadnienia oświetlania scen.
2. Zgłębić zasadę działania przykładowego modelu oświetlenia – Phonga.
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...

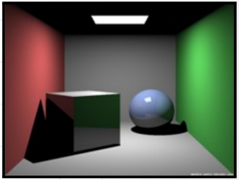
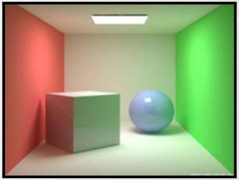
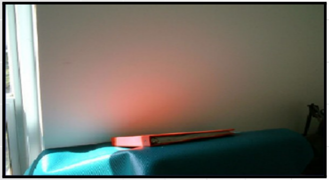




Źródło grafiki: https://pl.wikipedia.org/wiki/%C5%9Awiat%C5%82o_wolumetryczne
https://en.wikipedia.org/wiki/Volumetric_lighting

Rodzaje oświetlenia

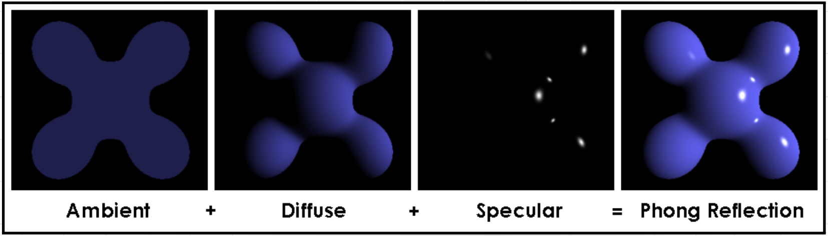
- Oświetlenie lokalne:
  - uwzględnia tylko bezpośredni wpływ źródła światła,
  - może uwzględniać elementy oświetlenia globalnego,
  - realizowane przez uproszczone modele oświetlenia.
- Oświetlenie globalne:
  - uwzględnia interakcje pomiędzy obiektami,
  - rzucanie cienia, załamania, wielokrotne odbicia,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

Źródło obrazów: https://slideplayer.com/slide/6399743/

Model Phonga

- 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.



Źródło: https://en.wikipedia.org/wiki/Phong_reflection_model

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

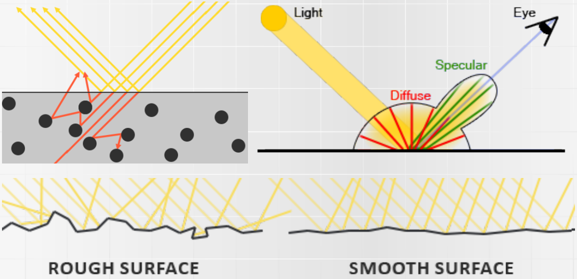
- Faktycznych parametrów w modelu jest 14.

Model Phonga – 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.

Inne modele oświetlenia

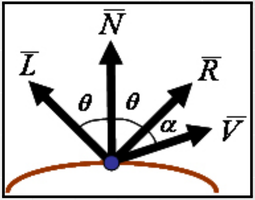
- Model Cook-Torrence'a,
  - http://www.codinglabs.net/article_physically_based_rendering_cook_torrance.aspx.
- Model Ashikhmina – Shirleya,
  - https://www.researchgate.net/publication/2523821_An_Anisotropic_Phong_Light_Reflection_Model.
- Hasło ogólne: **physically based rendering**,
  - https://learnopengl.com/PBR/Theory.



Źródła grafik: https://learnopengl.com/PBR/Theory
http://www.codinglabs.net/article_physically_based_rendering_cook_torrance.aspx

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.



Źródło: http://www.zsk.ict.pwr.wroc.pl/zsk/dyd/intinz/gk/lab/cw_5_dz/

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 Phonga.
- 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.

```python
def startup():
    ...

    glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient)
    glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse)
    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular)
    glMaterialf(GL_FRONT, GL_SHININESS, mat_shininess)

    glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient)
    glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse)
    glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular)
    glLightfv(GL_LIGHT0, GL_POSITION, light_position)

    glLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION, att_constant)
    glLightf(GL_LIGHT0, GL_LINEAR_ATTENUATION, att_linear)
    glLightf(GL_LIGHT0, GL_QUADRATIC_ATTENUATION, att_quadratic)

    glShadeModel(GL_SMOOTH)
    glEnable(GL_LIGHTING)
    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.

```
def render(time):
    global theta

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()

    gluLookAt(viewer[0], viewer[1], viewer[2],
              0.0, 0.0, 0.0, 0.0, 1.0, 0.0)

    if left_mouse_button_pressed:
        theta += delta_x * pix2angle

    glRotatef(theta, 0.0, 1.0, 0.0)

    quadric = gluNewQuadric()
    gluQuadricDrawStyle(quadric, GLU_FILL)
    gluSphere(quadric, 3.0, 10, 10)
    gluDeleteQuadric(quadric)

    glFlush()
```

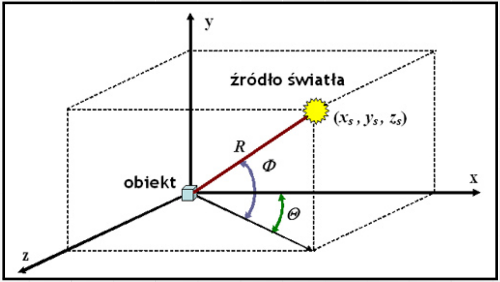
  – Warto zwrócić uwagę, iż z uwagi na określenie materiału, znaczenie tracą teraz wszystkie wywołania `glColor()` – ale nie trzeba ich usuwać z kodu.

Poruszanie źródłem światła dookoła obiektu

- Współrzędne źródła można określić za pomocą następujących równań,

$$
\begin{aligned}
x_s(R,\theta,\phi) &= R \cdot \cos(\theta) \cdot \cos(\phi)\,, \\
y_s(R,\theta,\phi) &= R \cdot \sin(\phi)\,, \\
z_s(R,\theta,\phi) &= R \cdot \sin(\theta) \cdot \cos(\phi)\,.
\end{aligned}
$$

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



Źródło obrazka: http://www.zsk.ict.pwr.wroc.pl/zsk/dyd/intinz/gk/lab/cw_5_dz/

Wektory normalne modelu z poprzednich zajęć (1/2)

- Współrzędne wierzchołków jajka można było obliczyć z układu równań,

$$x(u, v) = \left(-90 \cdot u^5 + 225 \cdot u^4 - 270 \cdot u^3 + 180 \cdot u^2 - 45 \cdot u\right) \cdot \cos\left(\pi \cdot v\right),$$
$$y(u, v) = 160 \cdot u^4 - 320 \cdot u^3 + 160 \cdot u^2 - 5,$$
$$z(u, v) = \left(-90 \cdot u^5 + 225 \cdot u^4 - 270 \cdot u^3 + 180 \cdot u^2 - 45 \cdot u\right) \cdot \sin\left(\pi \cdot v\right),$$

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,

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

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)

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

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

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

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

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

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

- 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`.

Politechnika Wrocławska

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 / w dół,
    - minimalna wartość składowej koloru to 0.0, a maksymalna 1.0,
    - pomocniczo – bieżące wartości można wypisywać w konsoli.

Politechnika Wrocławska

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,

```
quadric = gluNewQuadric()
gluQuadricDrawStyle(quadric, GLU_LINE)
gluSphere(quadric, 0.5, 6, 5)
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.