

OpenGL – oświetlenie

Bogdan Kreczmer

bogdan.kreczmer@pwr.wroc.pl

Katedra Cybernetyki i Robotyki
Wydziału Elektroniki
Politechnika Wroclawska

Kurs: Wizualizacja danych sensorycznych

Copyright©2017 Bogdan Kreczmer

Niniejszy dokument zawiera materiały do wykładu dotyczącego programowania obiektowego. Jest on udostępniony pod warunkiem wykorzystania wyłącznie do własnych prywatnych potrzeb i może on być kopiowany wyłącznie w całości, razem z niniejszą stroną tytułową.

Niniejsza prezentacja została wykonana przy użyciu systemu składu \LaTeX oraz stylu beamer, którego autorem jest Till Tantau.

Strona domowa projektu Beamer:

<http://latex-beamer.sourceforge.net>

- 1 Oświetlenie obiektów w OpenGL
 - Ogólna charakterystyka źródeł światła
 - Tworzenie i dodawanie oświetlenia

Rodzaje źródeł światła

Źródło światła może być punktowe. Jednak z takim źródłem stowarzyszone jest oświetlenie dookólne, które jest wynikiem wielokrotnych odbić światła od różnych powierzchni. Po wielu odbiciach światło to traci swoją pierwotną kierunkowość. W wyniku czego nie można określić jego pierwotnego pochodzenia.

Światło to jednak znika po wyłączeniu jego pierwotnego źródła.

Istnieje również ogólne rozproszone światło, które nie jest związane z żadnym źródłem.

Model oświetlenia

Model oświetlenia obejmuje 4 składowe, które wyznaczone są niezależnie, a następnie sumowane. Tymi składowymi są następujące typy oświetlenia:

dookólne (ang. ambient) – pochodzi z wielokrotnych odbić, które uniemożliwiają wyznaczenie jego pierwotnego źródła. Gdy tego typu oświetlenie pada na powierzchnię, odbijane jest we wszystkich kierunkach w jednakowym stopniu.

rozproszone (ang. diffuse) – światło docierające z jednego określonego kierunku. Ta składowa daje intensywniejsze oświetlenie, gdy światło pada na wprost na powierzchnię, niż pod pewnym kątem. Jednak od powierzchni odbija się w jednakowy sposób we wszystkich kierunkach.

zwierciadlane (ang. specular) – światło pochodzi je jednego określone kierunku. Po odbiciu od powierzchni ma tendencję rozchodzić się, w kierunku jednym kierunku. Powierzchnie mogą mieć różne współczynnika dla tego komponent. Metal może mieć bardzo wysoki współczynnik. Natomiast sukno może mieć współczynnik bliski zeru. Wartość tego współczynnika można interpretować jako połyskliwość.

Spektrum

Pomimo ustalonej charakterystyki widmowej źródła światła, składowe oświetlenia dookólnego, rozproszonego i zwierciadlanego mogą być różne.

Przykład: Jeśli jesteśmy w pokoju, którego ściany są zielone i jest ono oświetlone światłem białym, to składowa oświetlenia dookólnego będzie miała silniejszą składową zieloną.

Dla każdej składowej oświetlenia można indywidualnie ustawić komponenty RGB.

Kolory materiałów

Materiały mogą mieć różne komponenty spektralne dla dla składowych światła *ambient*, *diffuse* i *specular*.

Współczynniki tych komponentów decydują w jakim stopniu komponenty spektralne poszczególnych składowych światła zostaną odbite.

Dodatkowo materiały mają składową emisyjną (*ang. emissive*), która symuluje światło wydobywające się z danej powierzchni. Składowa ta pozwala zwiększyć intensywność koloru obiektu. Nie wpływa na nią żadne zewnętrzne oświetlenie, jak też nie daje ono żadnego dodatkowego oświetlenia otoczenia.

Wyznaczanie światła odbitego

$$(L_r, L_g, L_b) = (S_r \cdot M_r, S_g \cdot M_g, S_b \cdot M_b)$$

Intensywność światła pochodzącego z kilku źródeł lub odbić podlega sumowaniu

$$(L_r, L_g, L_b) = (L_{1,r} + L_{2,r}, L_{1,g} + L_{2,g}, L_{1,b} + L_{2,b})$$

Tworzenie oświetlenia

- Zdefiniuj normalne wektory dla każdego wierzchołka wszystkich obiektów. Wektory definiują orientację obiektu względem źródła światła.
- Stwórz, wyselekcjonuj i ustaw w odpowiedniej pozycji jedno lub więcej źródeł światła.
- Stwórz i wybierz model oświetlenia, który zdefiniuje poziom globalnego dookólnego oświetlenia i lokalizację punktu obserwacji (aby umożliwić wyliczenie efektu oświetlenia).
- Zdefiniuj własności materiałów obiektów znajdujących się na scenie.

Tworzenie oświetlenia – przykład

```
void init(void)
{
    GLfloat mat_specular[] = { 1.0, 1.0, 1.0, 1.0 };
    GLfloat mat_shininess[] = { 50.0 };
    GLfloat light_position[] = { 1.0, 1.0, 1.0, 0.0 };
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glShadeModel (GL_SMOOTH);

    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
    glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);
    glLightfv(GL_LIGHT0, GL_POSITION, light_position);

    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    glEnable(GL_DEPTH_TEST);
}

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glutSolidSphere(1.0, 20, 16);
    glFlush();
}
```

Tworzenie oświetlenia – Wektory normalne

```
void init(void)
{
    . . . .
}

void display(void)
{
    glClearColor(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glutSolidSphere(1.0, 20, 16);
    glFlush();
}
```

- Zdefiniuj normalne wektory dla każdego wierzchołka wszystkich obiektów. Wektory definiują orientację obiektu względem źródła światła.
- Stwórz, wyselekcjonuj i ustaw w odpowiedniej pozycji jedno lub więcej źródeł światła.
- Stwórz i wybierz model oświetlenia, który zdefiniuje poziom globalnego dookólnego oświetlenia i lokalizację punktu obserwacji (aby umożliwić wyliczenie efektu oświetlenia).
- Zdefiniuj własności materiałów obiektów znajdujących się na scenie.

Tworzenie oświetlenia – Wektory normalne

```
void init(void)
{
    . . . .
}

void display(void)
{
    glClearColor(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glutSolidSphere(1.0, 20, 16);
    glFlush();
}
```

- Zdefiniuj normalne wektory dla każdego wierzchołka wszystkich obiektów. Wektory definiują orientację obiektu względem źródła światła.

GLUT – The OpenGL Utility Toolkit

Niezależny od platformy toolkit umożliwiający pisanie programów wykorzystujących OpenGL. Implementuje prosty interfejs aplikacji dla *okienkowych* programów. Przewidziany jest do pisania małych lub średnich aplikacji.

Tworzenie oświetlenia – Źródło światła

```
void init(void)
{
    . . .
    GLfloat light_position [] = { 1.0, 1.0, 1.0, 0.0 };
    . . .
    glLightfv(GL_LIGHT0, GL_POSITION, light_position);

    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    . . .
}
```

- Zdefiniuj normalne wektory dla każdego wierzchołka wszystkich obiektów. Wektory definiują orientację obiektu względem źródła światła.
- **Stwórz, wyselekcjonuj i ustaw w odpowiedniej pozycji jedno lub więcej źródeł światła.**
- Stwórz i wybierz model oświetlenia, który zdefiniuje poziom globalnego dookólnego oświetlenia i lokalizację punktu obserwacji (aby umożliwić wyliczenie efektu oświetlenia).
- Zdefiniuj własności materiałów obiektów znajdujących się na scenie.

Tworzenie oświetlenia – Źródło światła

```
void init(void)
{
    . . .
    GLfloat light_position [] = { 1.0, 1.0, 1.0, 0.0 };
    . . .
    glLightfv(GL_LIGHT0, GL_POSITION, light_position);
    . . .
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    . . .
}
```

- Stwórz, wyselekcjonuj i ustaw w odpowiedniej pozycji jedno lub więcej źródeł światła.

Użyte zostało tylko jedno źródło światła, które domyślnie ma kolor biały. W ogólnym przypadku może być 8 źródeł o dowolnych barwach. Domyślnie pozostałe źródła mają kolor światła czarny. Należy pamiętać, że włączanie kolejnych źródeł zwiększa też ilość obliczeń. `glEnable(GL_LIGHTING)` przygotowuje OpenGL do wykonania obliczeń oświetlenia.

Tworzenie oświetlenia – Model oświetlenia

```
void init(void)
{
    . . .
    GLfloat light_position [] = { 1.0, 1.0, 1.0, 0.0 };
    . . .
    glLightfv(GL_LIGHT0, GL_POSITION, light_position);

    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    . . .
}
```

- Zdefiniuj normalne wektory dla każdego wierzchołka wszystkich obiektów. Wektory definiują orientację obiektu względem źródła światła.
- Stwórz, wyselekcjonuj i ustaw w odpowiedniej pozycji jedno lub więcej źródeł światła.
- Stwórz i wybierz model oświetlenia, który zdefiniuje poziom globalnego dookólnego oświetlenia i lokalizację punktu obserwacji (aby umożliwić wyliczenie efektu oświetlenia).
- Zdefiniuj własności materiałów obiektów znajdujących się na scenie.

Tworzenie oświetlenia – Model oświetlenia

```
void init(void)
{
    . . .
    GLfloat light_position[] = { 1.0, 1.0, 1.0, 0.0 };
    . . .
    glLightfv(GL_LIGHT0, GL_POSITION, light_position);

    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    . . .
}
```

- Stwórz i wybierz model oświetlenia, który zdefiniuje poziom globalnego dookólnego oświetlenia i lokalizację punktu obserwacji (aby umożliwić wyliczenie efektu oświetlenia).

Model oświetlenia definiowany jest przez funkcję `glLightModel*()`. W tym przykładzie zostało zdefiniowane tylko oświetlenie dookólne. Ogólnie model oświetlenia definiuje, czy oświetlenie ma być liczone dla frontu powierzchni i jej tyłu, czy źródło światła znajduje się w nieskończoności, czy też w określonej odległości. W przykładzie obliczenia są wykonywane tylko dla frontu, a źródło znajduje się w nieskończoności.

Tworzenie oświetlenia – Własności materiału

```
void init(void)
{
    GLfloat mat_specular[] = { 1.0, 1.0, 1.0, 1.0 }; /* RGBA */
    GLfloat mat_shininess[] = { 50.0 };
    . . .
    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
    glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);
    . . .
}
```

- Zdefiniuj normalne wektory dla każdego wierzchołka wszystkich obiektów. Wektory definiują orientację obiektu względem źródła światła.
- Stwórz, wyselekcjonuj i ustaw w odpowiedniej pozycji jedno lub więcej źródeł światła.
- Stwórz i wybierz model oświetlenia, który zdefiniuje poziom globalnego dookólnego oświetlenia i lokalizację punktu obserwacji (aby umożliwić wyliczenie efektu oświetlenia).
- Zdefiniuj własności materiałów obiektów znajdujących się na scenie.

Tworzenie oświetlenia – Własności materiału

```
void init(void)
{
    GLfloat mat_specular[] = { 1.0, 1.0, 1.0, 1.0 }; /* RGBA */
    GLfloat mat_shininess[] = { 50.0 };
    . . .
    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
    glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);
    . . .
}
```

- Zdefiniuj własności materiałów obiektów znajdujących się na scenie.

Rodzaj materiału definiuje sposób odbicia światła. Ponieważ interakcja różnych sposobów odbicia światła jest bardzo złożona, dobór właściwych parametrów jest sztuką samą w sobie. Ogólnie można określić dla powierzchni można określić takie parametry jak: *ambient*, *diffuse* oraz *specular*; niezależnie dla każdej z barw RGB. W prezentowanym przykładzie tylko dwie ostatnie własności zostały ustawione.

Tworzenie oświetlenia – Własności materiału

Prototypy

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

Nazwa parametru	Domyślna wartość	Znaczenie
GL_AMBIENT	(0.0, 0.0, 0.0, 1.0)	RGBA (intensywność światła)
GL_DIFFUSE	(1.0, 1.0, 1.0, 1.0)	RGBA (intensywność światła)
GL_SPECULAR	(1.0, 1.0, 1.0, 1.0)	RGBA (intensywność światła)
GL_POSITION	(0.0, 0.0, 1.0, 0.0)	(x, y, z, w) (pozycja światła)
GL_SPOT_DIRECTION	(0.0, 0.0, -1.0)	(x, y, z) (kierunek snopu światła)
GL_SPOT_EXPONENT	0.0	współczynnik eksponenty
GL_SPOT_CUTOFF	180.0	ograniczenie snopu światła
GL_CONSTANT_ATTENUATION	1.0	Stała zanikania
GL_LINEAR_ATTENUATION	0.0	Współczynnik liniowego zanikania
GL_QUADRATIC_ATTENUATION	0.0	Współczynnik zanikania z kwadratem odległości

Koniec prezentacji
Dziękuję za uwagę