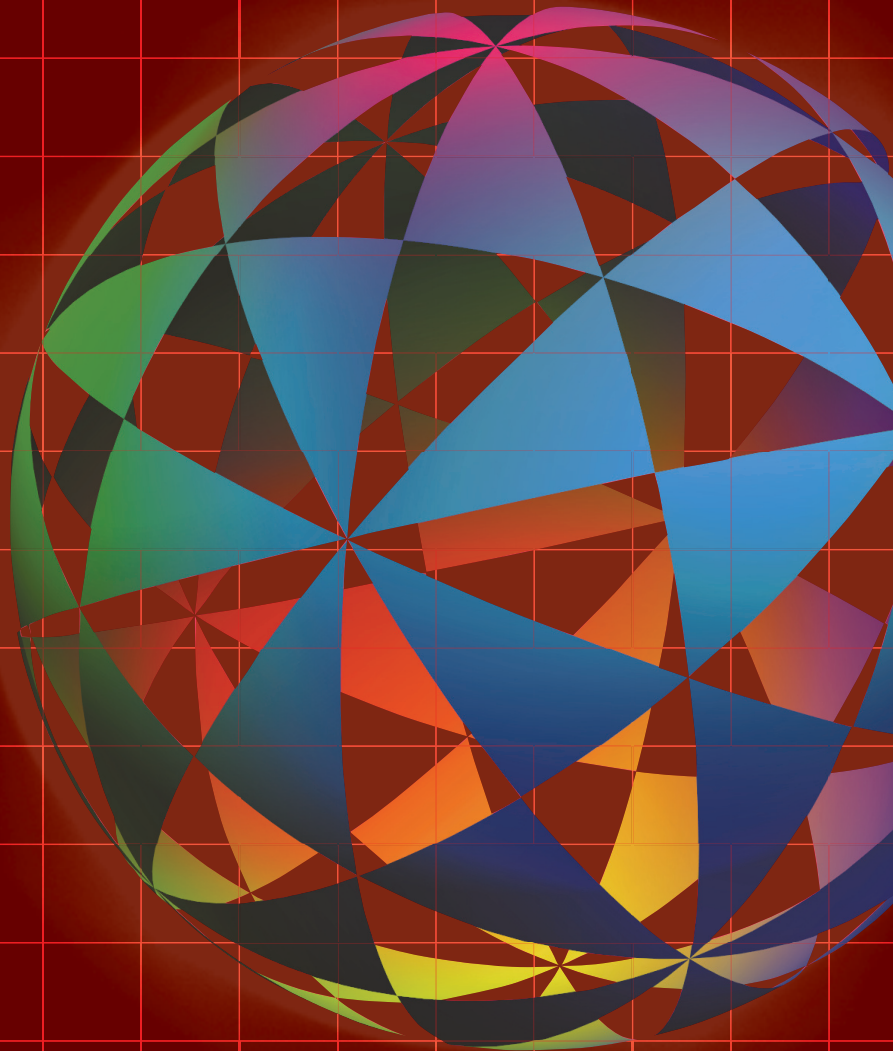


PRZEMYSŁAW KICIAK

OpenGL i GLSL

(nie taki krótki kurs)

Część I



OpenGL i GLSL

Pamięci Michała Jankowskiego

PRZEMYSŁAW KICIAK

OpenGL i GLSL

(nie taki krótki kurs)

Część I

Projekt okładki i stron tytułowych ANNA LUDWICKA
Wydawca WIOLETA SZCZYGIELSKA-DYBCIAK
Redaktor prowadzący JOLANTA KOWALCZUK
Redaktor MARIA KASPERSKA
Skład systemem T_EX PRZEMYSŁAW KICIAK
Koordynator produkcji ANNA BĄCZKOWSKA

Recenzenci

prof. dr hab. KRZYSZTOF DIKS
prof. dr hab. inż. PRZEMYSŁAW ROKITA

Zastrzeżonych nazw firm i produktów użyto w książce wyłącznie w celu identyfikacji.

Książka, którą nabyłeś, jest dziełem twórcy i wydawcy. Prosimy, abyś przestrzegał praw, jakie im przysługują. Jej zawartość możesz udostępnić nieodpłatnie osobom bliskim lub osobiście znanym. Ale nie publikuj jej w Internecie. Jeśli cytujesz jej fragmenty, nie zmieniaj ich treści i koniecznie zaznacz, czyje to dzieło. A kopiując jej część, rób to jedynie na użytek osobisty.

Szanujmy cudzą własność i prawo
Więcej na www.legalnakultura.pl
Polska Izba Książki

Copyright © by Wydawnictwo Naukowe PWN SA, Warszawa 2019

ISBN 978-83-01-20862-2 cz. I
ISBN 978-83-01-20866-0 cz. I–III

Wydanie I
Warszawa 2019

Wydawnictwo Naukowe PWN
02-460 Warszawa, ul. Gottlieba Daimlera 2
tel. (0-22) 69 54 321, faks 22 69 54 288
infolinia 801 33 33 88
e-mail: pwn@pwn.com.pl, reklama@pwn.pl
www.pwn.pl

Druk i oprawa: Partner Poligrafia Andrzej Kardasz

Spis treści

Część I

Przedmowa	1
Źródła	5
1. Wprowadzenie	7
1.1. Odrobina historii i ideologii	7
1.2. Potok przetwarzania grafiki	10
1.3. Programy szaderów	13
1.4. Źródła danych w potoku przetwarzania grafiki	14
2. Biblioteki i pliki nagłówkowe OpenGL-a	17
2.1. Biblioteki pomocnicze GLEW i gl3w	19
2.2. Nazwy i typy danych w OpenGL-u	22
2.3. Przedrostki i przyrostki nazw procedur	22
2.4. Zestawienie bibliotek	23
3. Otoczenie OpenGL-a	25
3.1. Biblioteka FreeGLUT	25
3.1.1. FreeGLUT — uwagi dodatkowe	31
3.2. Biblioteka GLFW	31
3.2.1. GLFW — obsługa dżojstika	36
3.2.2. GLFW — uwagi dodatkowe	37
3.3. System X Window i biblioteka GLX	38
3.4. Uzupełnienia	47
3.4.1. Wymiary ekranu	47
3.4.2. *Technologia Optimus	48
4. Utensylia	51
4.1. Wypisanie informacji o wersji	51
4.2. Reakcje na błąd	51
4.3. Reprezentacje kodów źródłowych szaderów	52
4.4. Kompilowanie szaderów i łączenie programów	54
4.5. *Uzupełnienia	57
4.5.1. SPIR-V	57
4.5.2. Opcje programów szaderów	61

5.	Działania na wektorach i macierzach	63
5.1.	Działania na macierzach	63
5.2.	Punkty i wektory swobodne	65
5.3.	Współrzędne kartezjańskie, jednorodnie i barycentryczne	66
5.4.	Przekształcenia afiniczne	68
5.5.	Prostopadłość	71
5.6.	Orientacja	73
5.7.	Procedury	75
6.	Rzutowanie	87
6.1.	Klatka, aspekt ekranu i kostka standardowa	87
6.2.	Rzutowanie perspektywiczne	89
6.3.	Rzutowanie równoległe	92
6.4.	Przekształcenia rzutowanych wierzchołków	94
6.5.	Aksonometria	95
6.6.	Rzutowanie dla grafiki dwuwymiarowej	96
7.	Pierwsza aplikacja	97
7.1.	Szadery	97
7.2.	Przygotowanie programu szaderów	99
7.3.	Przygotowanie obiektów w programie	102
7.4.	Przekształcenia współrzędnych	102
7.5.	Tworzenie obiektu tablicy wierzchołków	105
7.6.	Rysowanie	109
7.7.	Interakcja	111
7.8.	Sprzątanie	112
7.9.	Uzupełnienia	114
7.10.	Ćwiczenia	116
8.	Aplikacja pierwsza A	119
8.1.	Składanie obrotów	119
8.2.	Obracanie obserwatora wokół obiektu	120
8.3.	Animacja	126
8.4.	Ćwiczenia	131
9.	Podstawy języka GLSL	133
9.1.	Symbole leksykalne	133
9.2.	Preprocesor	134
9.3.	Podstawowe typy zmiennych	135
9.3.1.	Typy wektorowe i macierzowe	136
9.3.2.	Struktury	137
9.3.3.	Tablice	137
9.4.	Deklaracje zmiennych	138
9.5.	Wyrażenia	139
9.6.	Instrukcje	141
9.7.	Podprogramy	142
9.8.	Zmienne wskazujące podprogramy	143
9.9.	Równoległość i jednolitość obliczeń	145

9.10. Bloki zmiennych interfejsu	145
9.11. Komunikacja między szaderami	146
9.11.1. Zmienne wbudowane szadera wierzchołków	147
9.11.2. Zmienne wbudowane szadera sterowania rozdrabnianiem	147
9.11.3. Zmienne wbudowane szadera rozdrabniania	147
9.11.4. Zmienne wbudowane szadera geometrii	148
9.11.5. Zmienne wbudowane szadera fragmentów	149
9.12. Kwalifikatory układu	150
9.13. Funkcje i procedury wbudowane	153
9.13.1. Funkcje elementarne	153
9.13.2. Funkcje związane z potęgowaniem	154
9.13.3. Funkcje geometryczne	155
9.13.4. Funkcje związane z kątami	156
9.13.5. Funkcje macierzowe	157
9.13.6. Funkcje relacji wektorowych	157
9.13.7. Funkcje i procedury dla liczb całkowitych	158
9.13.8. Funkcje i procedury dla tekstur i obrazów	159
9.13.9. Liczniki niepodzielne i niepodzielne operacje na pamięci	162
9.14. Szadery obliczeniowe	162
9.14.1. Zmienne wbudowane szadera obliczeniowego	163
10. Aplikacja pierwsza B	165
10.1. Szadery — pierwszy program	165
10.2. Model oświetlenia i drugi program szaderów	166
10.3. Cztery procedury pomocnicze	174
10.4. Aplikacja pierwsza B	176
10.5. Procedury obsługi świateł	182
10.6. Uzupełnienia	185
10.6.1. Cieniowanie Gourauda i Phonga	185
10.6.2. Mgła	185
10.6.3. *Obcinanie i odrzucanie	186
10.7. Ćwiczenia	188
11. Aplikacja pierwsza C	189
11.1. Reprezentacja fontów	189
11.2. Szadery	190
11.3. Fonty i procedury wyświetlania tekstu	194
11.4. Aplikacja pierwsza C	201
11.5. Ćwiczenia	204
11.6. *Uzupełnienia	205
12. Aplikacja pierwsza D	209
12.1. Szadery i programy szaderów	209
12.2. Aplikacja pierwsza D	218
12.3. Ćwiczenia	223
12.4. *Uzupełnienia	225

13. Aplikacja pierwsza E	227
13.1. Czytanie pliku SMF	227
13.2. Zmiany w aplikacji	239
13.3. Ćwiczenia	244

Część II

14. Druga aplikacja	245
14.1. Płaty powierzchni Béziera	245
14.2. Wymierne płaty Béziera	249
14.3. Szadery	251
14.4. Procedury wprowadzania i rysowania płatów Béziera	262
14.5. Czajnik z Utah	267
14.6. Druga aplikacja	268
14.7. Uzupełnienia	276
15. Aplikacja druga A	279
15.1. Wyświetlanie siatek kontrolnych — szadery	279
15.2. Wyświetlanie siatek kontrolnych — procedury w C	282
15.3. Nowe i zmienione procedury aplikacji	282
15.4. Ćwiczenia	284
15.5. Uzupełnienia	285
16. Aplikacja druga B	289
16.1. Iloczyn sferyczny i powierzchnie obrotowe	289
16.2. Konstruowanie reprezentacji torusa	291
16.3. Zmiany w aplikacji	293
16.4. Ćwiczenia	298
16.5. *Uzupełnienia	299
17. Aplikacja druga C	303
17.1. Modele oświetlenia Phonga i Blinna–Phonga	303
17.2. Szadery	305
17.3. Zmiany w aplikacji	310
17.4. Uzupełnienia	316
17.4.1. Test nożyczek	316
17.4.2. Wczesne testy fragmentu	316
17.4.3. Oświetlenie hemisferyczne	317
17.5. Ćwiczenia	319
18. Aplikacja druga D	321
18.1. Mipmapping	322
18.2. Szadery	322
18.3. Czytanie i pisanie plików TIFF	327
18.4. Procedury przygotowania tekstur	330
18.5. Zmiany w aplikacji	332
18.6. Antyaliasing	335

18.7.	Ćwiczenia	336
18.8.	Uzupełnienia	337
18.8.1.	Rozszerzanie dziedziny tekstur	337
18.8.2.	Jednoczesne używanie wielu tekstur	338
18.8.3.	Skróty w OpenGL-u 4.5	339
18.8.4.	Rzutowe odwzorowanie dziedziny tekstury	340
18.8.5.	Antyaliasing w aplikacjach biblioteki FreeGLUT	342
19.	Aplikacja druga E	343
19.1.	Algebra z geometrią	343
19.2.	Tworzenie obrazów poza oknem	345
19.3.	Szadery	345
19.4.	Procedury obsługi lustra	348
19.5.	Zmiany w aplikacji	352
19.6.	Ćwiczenia	357
20.	Aplikacja druga F	359
20.1.	Wektor normalny zaburzonej powierzchni	359
20.2.	Szadery	362
20.3.	Zmiany w aplikacji	370
20.4.	Ćwiczenia	371
20.5.	*Uzupełnienia	371
20.5.1.	Anizotropowy model oświetlenia	371
20.5.2.	Antyaliasing tekstur proceduralnych	374
20.5.3.	Modyfikowanie współrzędnych tekstury odkształceń	375
21.	Aplikacja druga G	381
21.1.	Konstrukcja rzutowania sceny dla źródeł światła	382
21.2.	Szadery	387
21.3.	Tworzenie buforów ramki i tekstur dla obszarów cienia	396
21.4.	Zmiany w aplikacji	399
21.5.	Uzupełnienia	406
21.5.1.	Poprawianie błędów reprezentacji obszaru cienia	406
21.5.2.	Antyaliasing cienia	408
21.6.	Ćwiczenia	410
22.	Aplikacja druga H	411
22.1.	Łańcuchy kinematyczne	411
22.2.	Procedury obsługi łańcucha kinematycznego	416
22.3.	Szader obliczeniowy artykulacji	430
22.4.	Zmiany w aplikacji	433
22.5.	Ćwiczenia	444
23.	Aplikacja druga I	445
23.1.	Równania ruchu i reguły zachowania cząsteczek	445
23.2.	Szadery układu cząsteczek	448
23.3.	Generatory liczb i wektorów pseudolosowych	451
23.4.	Przygotowanie, symulacja i rysowanie układu cząsteczek	453

23.5.	Zmiany łańcucha kinematycznego	461
23.6.	Algorytm cieni dla mgły	463
23.7.	Pozostałe zmiany w aplikacji	469
23.8.	Ćwiczenia	470
23.9.	*Uzupełnienia	471
23.9.1.	Funkcje mieszające	471
23.9.2.	Odzorowanie buforów w przestrzeń adresową CPU	472
24.	Aplikacja druga J	473
24.1.	Podstawy symulacji głębi ostrości	473
24.2.	Implementacja bufora akumulacji	479
24.3.	Obliczanie parametrów rzutowania	486
24.4.	Dalsze zmiany w aplikacji	489
24.5.	Ćwiczenia	499
25.	Aplikacja druga K	501
25.1.	Rysowanie na wielu warstwach	501
25.2.	Stereoskopia	510
25.3.	Ćwiczenia	515
25.4.	Uzupełnienia	516
25.4.1.	Tekstury i obrazy	516
25.4.2.	Tekstury sześciánowe	518
25.4.3.	*Prymitywy z przyległościami	526
 Część III		
26.	Graficzny interfejs użytkownika w X Window	531
26.1.	Struktury danych i procedury podstawowe	532
26.2.	Procedury przekazujące komunikaty	537
26.3.	Kodowanie kolorów w systemie X Window	542
26.4.	Przykłady wihajstrów	544
27.	Zagęszczanie siatek	551
27.1.	Definicja i warunki poprawności siatki	551
27.2.	Reprezentacja siatki w pamięci RAM CPU	553
27.3.	Reprezentacja siatki w pamięci GPU	554
27.4.	Podwajanie i uśrednianie siatki	559
27.5.	Obliczanie sum prefiksowych	561
27.6.	Zmienne szadera zagęszczenia siatek	565
27.7.	Implementacja podwajania	568
27.8.	Implementacja uśredniania	583
27.9.	Procedura zagęszczenia siatki	592
27.10.	Przygotowanie i likwidacja programu zagęszczenia siatek	593
27.11.	Ćwiczenia	594
28.	Trzecia aplikacja	595
28.1.	Model dłóni	595

28.2.	Rysowanie siatki	596
28.3.	Okna trzeciej aplikacji	604
28.4.	Obsługa przekształceń	617
28.5.	Wyświetlane obiekty	619
28.6.	Ćwiczenia	621
29.	Aplikacja trzecia A	623
29.1.	Obliczanie wektorów normalnych	623
29.2.	Rysowanie siatki	629
29.3.	Zmiany w aplikacji	634
29.4.	Ćwiczenia	636
30.	Aplikacja trzecia B	637
30.1.	Łańcuch kinematyczny	637
30.2.	Rysowanie sceny	646
30.3.	Interfejs użytkownika	646
30.4.	Pozostałe zmiany w aplikacji	648
30.5.	Ćwiczenia	650
31.	Aplikacja trzecia C	651
31.1.	Łańcuch kinematyczny	651
31.2.	Szadery i procedury rysowania sceny	660
31.3.	Pozostałe zmiany w aplikacji	669
31.4.	Ćwiczenia	670
31.5.	Uzupełnienia	671
32.	Aplikacja trzecia D	673
32.1.	Działanie interfejsu użytkownika	673
32.2.	Wihajster osi czasu	675
32.3.	Procedury obsługi animacji	685
32.4.	Menu trzeciego podokna	694
32.5.	Pozostałe zmiany w aplikacji	698
32.6.	*Ćwiczenia	698
A.	Jeszcze trochę algebry z geometrią	699
A.1.	Załamane światła	699
A.2.	Konstrukcja obrotu do ustalonego położenia	700
A.3.	Rozkładanie przekształceń afinicznych	702
A.4.	Kwaterniony i obroty	706
B.	Krzywe i powierzchnie B-sklejane	717
B.1.	Określenie funkcji, krzywych i płatów B-sklejanych	717
B.2.	Algorytmy de Boora	719
B.3.	B-sklejane krzywe interpolacyjne	725
B.4.	Sklejane krzywe kwaternionowe	731
C.	Światło, kolory, barwy i ich współrzędne	735
C.1.	Radiometria i fotometria	735

C.2.	Widzenie trójbarwne	740
C.3.	Diagram CIE	742
C.4.	Układy współrzędnych RGB	745
C.5.	Układy z luminancją i chrominancją	747
C.6.	Układy z subtraktywnym mieszaniem barw	748
C.7.	Układy HSV i HSL	749
D.	Dźojstik w aplikacjach X Window	751
D.1.	Aktywne sprawdzanie	751
D.2.	Komunikacja za pośrednictwem systemu X Window	756
E.	Rzutowanie nieliniowe	763
E.1.	Panorama punktowa	763
E.2.	Panorama linearna	765
E.3.	Rzutowanie na sferę	766
E.4.	Rozdrabnianie w rzutowaniu nieliniowym	767
F.	Słowniki	775
F.1.	Słownik TLS-ów i CzLS-ów	775
F.2.	Słownik wyrazów wieloznacznych	779
	Skorowidz	783

Spis treści części I

Przedmowa	1
Źródła	5
1. Wprowadzenie	7
1.1. Odrobina historii i ideologii	7
1.2. Potok przetwarzania grafiki	10
1.3. Programy szaderów	13
1.4. Źródła danych w potoku przetwarzania grafiki	14
2. Biblioteki i pliki nagłówkowe OpenGL-a	17
2.1. Biblioteki pomocnicze GLEW i gl3w	19
2.2. Nazwy i typy danych w OpenGL-u	22
2.3. Przedrostki i przyrostki nazw procedur	22
2.4. Zestawienie bibliotek	23
3. Otoczenie OpenGL-a	25
3.1. Biblioteka FreeGLUT	25
3.1.1. FreeGLUT — uwagi dodatkowe	31
3.2. Biblioteka GLFW	31
3.2.1. GLFW — obsługa dżojstika	36
3.2.2. GLFW — uwagi dodatkowe	37
3.3. System X Window i biblioteka GLX	38
3.4. Uzupełnienia	47
3.4.1. Wymiary ekranu	47
3.4.2. *Technologia Optimus	48
4. Utensylia	51
4.1. Wypisanie informacji o wersji	51
4.2. Reakcje na błąd	51
4.3. Reprezentacje kodów źródłowych szaderów	52
4.4. Kompilowanie szaderów i łączenie programów	54
4.5. *Uzupełnienia	57
4.5.1. SPIR-V	57
4.5.2. Opcje programów szaderów	61

5.	Działania na wektorach i macierzach	63
5.1.	Działania na macierzach	63
5.2.	Punkty i wektory swobodne	65
5.3.	Współrzędne kartezjańskie, jednorodnie i barycentryczne	66
5.4.	Przekształcenia afiniczne	68
5.5.	Prostopadłość	71
5.6.	Orientacja	73
5.7.	Procedury	75
6.	Rzutowanie	87
6.1.	Klatka, aspekt ekranu i kostka standardowa	87
6.2.	Rzutowanie perspektywiczne	89
6.3.	Rzutowanie równoległe	92
6.4.	Przekształcenia rzutowanych wierzchołków	94
6.5.	Aksonometria	95
6.6.	Rzutowanie dla grafiki dwuwymiarowej	96
7.	Pierwsza aplikacja	97
7.1.	Szadery	97
7.2.	Przygotowanie programu szaderów	99
7.3.	Przygotowanie obiektów w programie	102
7.4.	Przekształcenia współrzędnych	102
7.5.	Tworzenie obiektu tablicy wierzchołków	105
7.6.	Rysowanie	109
7.7.	Interakcja	111
7.8.	Sprzątanie	112
7.9.	Uzupełnienia	114
7.10.	Ćwiczenia	116
8.	Aplikacja pierwsza A	119
8.1.	Składanie obrotów	119
8.2.	Obracanie obserwatora wokół obiektu	120
8.3.	Animacja	126
8.4.	Ćwiczenia	131
9.	Podstawy języka GLSL	133
9.1.	Symbole leksykalne	133
9.2.	Preprocesor	134
9.3.	Podstawowe typy zmiennych	135
9.3.1.	Typy wektorowe i macierzowe	136
9.3.2.	Struktury	137
9.3.3.	Tablice	137
9.4.	Deklaracje zmiennych	138
9.5.	Wyrażenia	139
9.6.	Instrukcje	141
9.7.	Podprogramy	142
9.8.	Zmienne wskazujące podprogramy	143
9.9.	Równoległość i jednolitość obliczeń	145

9.10. Bloki zmiennych interfejsu	145
9.11. Komunikacja między szaderami	146
9.11.1. Zmienne wbudowane szadera wierzchołków	147
9.11.2. Zmienne wbudowane szadera sterowania rozdrabnianiem	147
9.11.3. Zmienne wbudowane szadera rozdrabniania	147
9.11.4. Zmienne wbudowane szadera geometrii	148
9.11.5. Zmienne wbudowane szadera fragmentów	149
9.12. Kwalifikatory układu	150
9.13. Funkcje i procedury wbudowane	153
9.13.1. Funkcje elementarne	153
9.13.2. Funkcje związane z potęgowaniem	154
9.13.3. Funkcje geometryczne	155
9.13.4. Funkcje związane z kątami	156
9.13.5. Funkcje macierzowe	157
9.13.6. Funkcje relacji wektorowych	157
9.13.7. Funkcje i procedury dla liczb całkowitych	158
9.13.8. Funkcje i procedury dla tekstur i obrazów	159
9.13.9. Liczniki niepodzielne i niepodzielne operacje na pamięci	162
9.14. Szadery obliczeniowe	162
9.14.1. Zmienne wbudowane szadera obliczeniowego	163
10. Aplikacja pierwsza B	165
10.1. Szadery — pierwszy program	165
10.2. Model oświetlenia i drugi program szaderów	166
10.3. Cztery procedury pomocnicze	174
10.4. Aplikacja pierwsza B	176
10.5. Procedury obsługi świateł	182
10.6. Uzupełnienia	185
10.6.1. Cieniowanie Gourauda i Phonga	185
10.6.2. Mgła	185
10.6.3. *Obcinanie i odrzucanie	186
10.7. Ćwiczenia	188
11. Aplikacja pierwsza C	189
11.1. Reprezentacja fontów	189
11.2. Szadery	190
11.3. Fonty i procedury wyświetlania tekstu	194
11.4. Aplikacja pierwsza C	201
11.5. Ćwiczenia	204
11.6. *Uzupełnienia	205
12. Aplikacja pierwsza D	209
12.1. Szadery i programy szaderów	209
12.2. Aplikacja pierwsza D	218
12.3. Ćwiczenia	223
12.4. *Uzupełnienia	225

13. Aplikacja pierwsza E	227
13.1. Czytanie pliku SMF	227
13.2. Zmiany w aplikacji	239
13.3. Ćwiczenia	244

Powiedziała, że musimy dziś robić to, na co mamy ochotę.

TOVE JANSSON: *Dolina Muminków w listopadzie*¹

Przedmowa

Standard OpenGL opisuje sposób, w jaki programy mogą tworzyć obrazy obiektów trójwymiarowych z wykorzystaniem nowoczesnego sprzętu i jego mocy obliczeniowej. Jednocześnie standard ten umożliwia przenośność programów, które mogą działać na komputerach wyposażonych w procesory graficzne różnych typów (i pochodzące od różnych producentów).

Od swojego powstania, już ponad ćwierć wieku temu, OpenGL udostępnia dla programów pełne możliwości sprzętu. Ścisłej biorąc, kolejne wersje OpenGL-a są rozwijane razem z procesorami graficznymi, aby umożliwić korzystanie z ich najnowszych cech, pozostając przy tym możliwie łatwym w użyciu interfejsem programowania aplikacji. Do roku 2006 zestaw „gotowych do użycia” algorytmów realizowanych przez procedury OpenGL-a był stale rozszerzany. Stanowiąc wygodne oparcie dla programistów, procedury te umożliwiały otrzymywanie coraz większego repertuaru efektów na obrazach, choć repertuar ten pozostawał ograniczony. W 2006 roku standard uległ zasadniczemu przeobrażeniu; znikło wtedy wiele ograniczeń, kosztem jednoczesnej rezygnacji ze wspomnianego wygodnego oparcia, jakie dawały „gotowe” algorytmy. Wprowadzona nieco wcześniej *możliwość* programowania procesorów graficznych sprawiła, że jedynym ograniczeniem podczas pisania programów pozostała wyobraźnia (i budżet) programisty, ale aby to osiągnąć, trzeba było zastąpić większość tych algorytmów programami, które autor aplikacji *musi* dostarczyć. Do ich pisania został stworzony język GLSL — podobny do C język „wysokiego poziomu”, umożliwiający skupienie się na rozwiązywaniu zadaniu zamiast na zestawie rejestrów i rozkazów maszynowych procesora graficznego.

Pisania programów korzystających z OpenGL-a można się uczyć, korzystając z rozmaitych książek i (dostępnych w Internecie) specyfikacji. Jednak w moim przekonaniu obfitość materiału przedstawionego w istniejących podręcznikach jest ich wadą: autorzy starają się pokazać jak najwięcej elementów OpenGL-a, z konieczności ilustrując je fragmentarycznymi przykładami, co pozostawia u czytelników mieszane uczucia fascynacji, zagubienia i niedosytu (z dodatkiem frustracji podczas poszukiwania odpowiedzi na konkretne pytanie). Natomiast specyfikacje szczegółowo opisują procedury OpenGL-a, wyliczając *wszystkie* dopuszczalne wartości parametrów i *wszystkie* opcje, które można za ich pomocą włączyć albo wyłączyć, co jednak nie bardzo pomaga nowicjuszowi zgadnąć, jak poszczególne procedury

¹Przekład Teresy Chłapowskiej, Nasza Księgarnia, 1980.

współpracują ze sobą, w jakiej kolejności należy je wywoływać i jakie wartości parametrów trzeba podać w konkretnej sytuacji, aby osiągnąć pożądaný (albo w ogóle jakiś widoczny) efekt. W specyfikacjach nie ma też mowy o algorytmach i projektach, które warto by zrealizować za pomocą tych procedur.

Książka *OpenGL i GLSL (nie taki krótki kurs)* powstała ze skryptu *OpenGL i GLSL (krótki kurs)*, który napisałem na potrzeby ćwiczeń z grafiki komputerowej prowadzonych na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego. Kurs, o którym mówi tytuł, polega na zapoznaniu uczestnika z wybranymi² elementami standardu OpenGL i języka GLSL za pomocą kompletnych, działających przykładów. Celem tego kursu *nie jest* zatem przedstawienie wszystkich możliwości, opcji, szczegółów i niuansów OpenGL-a, a raczej podanie opisu jak najkrótszej drogi do osiągnięcia wyniku — programu, który wyświetla obrazy w oknie na ekranie i który działa w trybie interakcji czynnej, na bieżąco reagując na działania użytkownika mającego do dyspozycji mysz i klawiaturę. W początkowych rozdziałach zawarłem niezbędne informacje podstawowe, a w kolejnych przedstawiłem krok po kroku aplikacje napisane z wykorzystaniem różnych dostępnych w OpenGL-u konstrukcji. Każda aplikacja ma kilka wariantów; każdy kolejny wariant powstał *po uruchomieniu* wariantu poprzedniego przez dodanie czegoś nowego i ilustruje sposób używania kolejnych elementów OpenGL-a i języka GLSL. Nauka w ramach kursu polega na eksperymentowaniu z aplikacjami i samodzielnym wprowadzaniu zmian, na przykład na dodawaniu do aplikacji nowych obiektów do narysowania lub nowych efektów wizualnych określonych przez Czytelnika — kursanta.

Wielu autorów, usiłując przekonać swoich czytelników o potrzebie stosowania matematyki w grafice komputerowej, przedstawia ją (matematykę) niemal jak wroga, co czasami wygląda jak groteskowa kokieteria; tak odbieram tytuł podrozdziału „Czy to ten okrutny rozdział o matematyce?” z jednej oraz słowa *“we spare you the gory detail”* z innej będącej w moim posiadaniu książki o OpenGL-u (*mimo to* obie te książki uważam za pożyteczne, wiele się z nich dowiedziałem i nauczyłem). Swoje zdanie w tej sprawie ujmę następująco: co najmniej tak samo ważne jak sposób przesyłania macierzy dowolnego przekształcenia do pamięci karty graficznej jest wcześniejsze obliczenie współczynników tej macierzy oparte na poprawnej matematycznie konstrukcji. Matematyka użyta do napisania aplikacji uruchomionych w ramach mojego kursu nie powinna sprawić najmniejszych kłopotów studentom, którzy zaliczyli na pierwszym roku Algebrę Liniową i Analizę, a dla Czytelników, którzy tych przedmiotów nie studiowali, właśnie grafika komputerowa jest najlepszą okazją do poznania tej odrobiny dosyć łatwej (i niewiele wykraczającej poza program szkoły średniej) matematyki, bez której nie ma nie tylko grafiki, ale i mnóstwa innych sposobów radzenia sobie z różnymi problemami przy użyciu komputerów. W tej książce nie ma twierdzeń ani formalnych dowodów, ale potrzebne do napisania aplikacji wzory są porządnie wyprowadzone na podstawie dobrze znanych faktów, a przybliżenia lub nieścisłości w rozumowaniach wyraźnie zaznaczyłem, bo tak trzeba. Nieco bardziej zaawansowana (ale też nietrudna) matematyka opisana w dodatkach A, B i C przyda się Czytelnikom w ich własnych projektach.

²przeze mnie, całkowicie arbitralnie

Ćwiczenie. Przeczytaj głośno zdanie:

Poznam odrobinę matematyki i nie zawaham się jej użyć!

Czytelnik tej książki nie dowie się, gdzie w moim ulubionym zintegrowanym środowisku programistycznym należy kliknąć, aby utworzyć nowy pusty projekt, i to z dwóch powodów. Po pierwsze, nie odmawiając wartości zintegrowanym środowiskom programistycznym, nie używam żadnego z nich, dzięki czemu mogę wszystkich Czytelników traktować, cokolwiek to znaczy, sprawiedliwie. Po drugie, słowo „kliknąć” uważam w najlepszym razie za chwast, a w najgorszym za nowotwór na języku polskim i nie używam takich wyrazów. Zdając sobie sprawę, że walka z wywołującymi ból uszu i przygnębienie słowami — potworkami w rodzaju „mapowanie”, „renderowanie”, „werteksy” czy „rejkasty” (!) może być już nie do wygrania, podjąłem tutaj próbę ustalenia terminologii szanującej zarówno sens nazywanych pojęć, jak i język, którym mam przywilej i zaszczyt posługiwać się codziennie. Ale w zbiorze znaków dopuszczalnych w programach w C nie ma wszystkich liter polskiego alfabetu, a ponadto mało co wygląda tak źle jak tekst z wymieszanymi (i zniekształconymi) słowami z dwóch języków. Dlatego szanując także język angielski, z którego pochodzą słowa kluczowe języków C i GLSL i nazwy procedur OpenGL-a i licząc na zrozumienie u Czytelników, w aplikacjach przedstawionych w tej książce znaczące identyfikatory (nazwy makrodefinicji, typów, zmiennych i procedur) utworzyłem ze słów angielskich. Ponadto, choć to zwyczaj anglosaski, liczby rzeczywiste w tekście zapisuję tak jak w programach, na przykład 0.5533, dla jednolitości notacji.

Każdy program komputerowy funkcjonuje w pewnym środowisku, tzn. w konkretnym systemie operacyjnym i podsystemie obsługującym urządzenia wejściowe i ekran. Aplikacje opisane w tej książce uruchomiłem w systemie Linux/X Window. Współpraca aplikacji ze środowiskiem może się odbywać za pośrednictwem biblioteki pomocniczej mającej własny interfejs programowania aplikacji, za którym biblioteka ta ukrywa środowisko. Korzystając z takiej biblioteki, można pisać aplikacje, których przeniesienie do innego systemu operacyjnego wymaga tylko ich skompilowania w tym systemie. W tej książce są przedstawione dwie takie biblioteki, FreeGLUT i GLFW. Związanie aplikacji z konkretnym systemem operacyjnym i podsystemem wejścia/wyjścia likwiduje jej przenośność, ale za to umożliwia pełniejsze wykorzystanie możliwości tego środowiska. Dlatego pokazałem także aplikacje, które współpracują bezpośrednio z systemem X Window, używając go do stworzenia (bardzo prostego, ale o nieograniczonych możliwościach rozbudowy) graficznego interfejsu użytkownika.

Podstawowy zamysł, z jakim zabrałem się do tworzenia aplikacji i ich opisów w skrypcie, a potem w książce, był taki, aby żadna linia napisanego przeze mnie kodu nie miała przed Czytelnikami tajemnic. Temu służy zarówno wybór języka programowania (ANSI C, a nie np. C++), jak i zamieszczenie na listingach i skomentowanie w tekście prawie całego kodu aplikacji (w tym również podprogramów pomocniczych, które nie wywołują procedur OpenGL-a). W dalszych rozdziałach opuściłem jednak długie na kilkaset linii ciągi liczb (np. współrzędnych punktów definiujących obiekty do narysowania lub bitowych wzorców znaków używanych do wyświetlania napisów), a ponadto pomiąłem fragmenty kodu realizujące rutynowe zadania przedstawione w opisach wcześniejszych aplikacji (takie jak kom-