

## **Przedmowa (13)**

## **Wstęp (15)**

## **Rozdział 1. Wprowadzenie (21)**

- 1.1. Czym jest Cg? (21)
  - o 1.1.1. Języki dla programowalnego sprzętu graficznego (22)
  - o 1.1.2. Model przepływu danych w Cg (22)
  - o 1.1.3. Specjalizacja a generalizacja procesorów graficznych (23)
  - o 1.1.4. Wydajność języka Cg (24)
  - o 1.1.5. Współdziałanie z konwencjonalnymi językami (24)
  - o 1.1.6. Inne aspekty języka Cg (26)
  - o 1.1.7. Ograniczone środowisko wykonywania programów Cg (27)
- 1.2. Wierzchołki, fragmenty i potok grafiki (28)
  - o 1.2.1. Ewolucja sprzętu graficznego (28)
  - o 1.2.2. Cztery generacje sprzętu graficznego (29)
  - o 1.2.3. Sprzętowy potok graficzny (33)
  - o 1.2.4. Programowalny potok graficzny (37)
  - o 1.2.5. Język Cg zapewnia możliwość programowania jednostek wierzchołków i fragmentów (40)
- 1.3. Historia powstania Cg (40)
  - o 1.3.1. Współpraca firm NVIDIA i Microsoft w celu określenia języków Cg i HLSL (42)
  - o 1.3.2. Nieinteraktywne języki cieniowania (42)
  - o 1.3.3. Interfejsy programistyczne w grafice trójwymiarowej (45)
- 1.4. Środowisko Cg (45)
  - o 1.4.1. Standardowe interfejsy programistyczne 3D: OpenGL i Direct3D (45)
  - o 1.4.2. Kompilator i biblioteka wykonywania Cg (47)
  - o 1.4.3. Narzędzia CgFX i format pliku (49)
- 1.5. Ćwiczenia (53)

## **Rozdział 2. Najprostsze programy (55)**

- 2.1. Prosty program wierzchołków (55)
  - o 2.1.1. Struktura wyjścia (56)
  - o 2.1.2. Identyfikatory (57)
  - o 2.1.3. Elementy struktur (58)
  - o 2.1.4. Wektory (58)
  - o 2.1.5. Macierze (58)
  - o 2.1.6. Semantyka (59)
  - o 2.1.7. Funkcje (60)
  - o 2.1.8. Różnice w semantyce wejścia i wyjścia (61)
  - o 2.1.9. Ciało funkcji (62)
- 2.2. Kompilacja przykładu (64)
  - o 2.2.1. Profile programu wierzchołków (64)
  - o 2.2.2. Klasy błędów kompilacji programów Cg (66)
  - o 2.2.3. Błędy wynikające ze złego profilu (66)
  - o 2.2.4. Norma - kilka funkcji wejścia (68)
  - o 2.2.5. Pobieranie i konfiguracja programów wierzchołków i fragmentów (68)
- 2.3. Prosty program fragmentów (70)
  - o 2.3.1. Profile dla programów fragmentów (71)

- 2.4. Rendering przykładowych programów wierzchołków i fragmentów (72)
  - o 2.4.1. Rendering trójkąta w OpenGL (73)
  - o 2.4.2. Rendering trójkąta w Direct3D (74)
  - o 2.4.3. Uzyskanie tych samych wyników (74)
- 2.5. Ćwiczenia (76)

### **Rozdział 3. Parametry, tekstury i wyrażenia (77)**

- 3.1. Parametry (77)
  - o 3.1.1. Parametry jednolite (77)
  - o 3.1.2. Kwalifikator typu const (80)
  - o 3.1.3. Różnorodność parametrów (80)
- 3.2. Próbkowanie tekstur (82)
  - o 3.2.1. Obiekty próbek (82)
  - o 3.2.2. Próbkowanie tekstur (83)
  - o 3.2.3. Wysyłanie współrzędnych tekstury w trakcie próbkowania tekstury (84)
- 3.3. Wyrażenia matematyczne (85)
  - o 3.3.1. Operatory (85)
  - o 3.3.2. Typy danych uzależnione od profilu (86)
  - o 3.3.3. Funkcje wbudowane w standardową bibliotekę Cg (90)
  - o 3.3.4. Skręcanie w dwuwymiarze (93)
  - o 3.3.5. Efekt podwójnego widzenia (96)
- 3.4. Ćwiczenia (100)

### **Rozdział 4. Przekształcenia (101)**

- 4.1. Układy współrzędnych (101)
  - o 4.1.1. Przestrzeń obiektu (102)
  - o 4.1.2. Współrzędne homogeniczne (103)
  - o 4.1.3. Przestrzeń świata (103)
  - o 4.1.4. Przekształcenie modelu (104)
  - o 4.1.5. Przestrzeń oka (105)
  - o 4.1.6. Przekształcenie widoku (105)
  - o 4.1.7. Przestrzeń przycięcia (106)
  - o 4.1.8. Przekształcenie rzutowania (106)
  - o 4.1.9. Znormalizowane współrzędne urządzenia (107)
  - o 4.1.10. Współrzędne okna (108)
- 4.2. Zastosowanie teorii (108)
- 4.3. Ćwiczenia (109)

### **Rozdział 5. Oświetlenie (111)**

- 5.1. Oświetlenie i związane z nim modele (111)
- 5.2. Implementacja podstawowego modelu oświetlenia opartego na wierzchołkach (113)
  - o 5.2.1. Podstawowy model oświetlenia (113)
  - o 5.2.2. Program wierzchołków dla prostego oświetlenia opartego na wierzchołkach (119)
  - o 5.2.3. Program fragmentów dla modelu oświetlenia wykorzystującego wierzchołki (128)

- o 5.2.4. Efekt modelu oświetlenia opartego na wierzchołkach (128)
- 5.3. Model oświetlenia oparty na fragmentach (129)
  - o 5.3.1. Implementacja modelu oświetlenia opartego na fragmentach (130)
  - o 5.3.2. Program wierzchołków dla modelu oświetlenia opartego na fragmentach (131)
  - o 5.3.3. Program fragmentów dla modelu oświetlenia opartego na fragmentach (131)
- 5.4. Tworzenie funkcji modelu oświetlenia (133)
  - o 5.4.1. Deklarowanie funkcji (133)
  - o 5.4.2. Funkcja oświetlenia (134)
  - o 5.4.3. Struktury (135)
  - o 5.4.4. Tablice (136)
  - o 5.4.5. Sterowanie wykonywaniem programu (137)
  - o 5.4.6. Obliczenie modelu oświetlenia rozproszenia i rozbłysku (138)
- 5.5. Rozszerzenie modelu podstawowego (138)
  - o 5.5.1. Zanik światła wraz z odległością (139)
  - o 5.5.2. Dodanie efektu reflektora (140)
  - o 5.5.3. Światła kierunkowe (145)
- 5.6. Ćwiczenia (145)

## **Rozdział 6. Animacja (147)**

- 6.1. Ruch w czasie (147)
- 6.2. Pulsujący obiekt (148)
  - o 6.2.1. Program wierzchołków (149)
  - o 6.2.2. Obliczanie przemieszczenia (150)
- 6.3. Systemy cząsteczek (152)
  - o 6.3.1. Warunki początkowe (153)
  - o 6.3.2. Wektoryzacja obliczeń (153)
  - o 6.3.3. Parametry systemu cząsteczek (154)
  - o 6.3.4. Program wierzchołków (154)
  - o 6.3.5. Ubieramy system cząsteczek (156)
- 6.4. Interpolacja ujęć kluczowych (157)
  - o 6.4.1. Teoria ujęć kluczowych (157)
  - o 6.4.2. Rodzaje interpolacji (160)
  - o 6.4.3. Prosta interpolacja ujęć kluczowych (160)
  - o 6.4.4. Interpolacja ujęć kluczowych z oświetleniem (162)
- 6.5. System skóry dla wierzchołków (163)
  - o 6.5.1. Teoria systemu skóry dla wierzchołków (163)
  - o 6.5.2. System skóry w programie wierzchołków (166)
- 6.6. Ćwiczenia (167)

## **Rozdział 7. Mapowanie środowiska (169)**

- 7.1. Mapowanie środowiska (169)
  - o 7.1.1. Tekstury map sześciennych (170)
  - o 7.1.2. Generowanie map sześciennych (171)
  - o 7.1.3. Koncepcja mapowania środowiska (171)
  - o 7.1.4. Obliczenie wektorów odbicia (172)
  - o 7.1.5. Założenia mapowania środowiska (173)

- 7.2. Mapowanie odbić (174)
  - o 7.2.1. Parametry określone przez aplikację (175)
  - o 7.2.2. Program wierzchołków (175)
  - o 7.2.3. Program fragmentów (179)
  - o 7.2.4. Mapy sterujące (180)
  - o 7.2.5. Program wierzchołków a program fragmentów (180)
- 7.3. Mapowanie załamania (181)
  - o 7.3.1. Zjawisko załamania światła (182)
  - o 7.3.2. Program wierzchołków (184)
  - o 7.3.3. Program fragmentów (186)
- 7.4. Efekt Fresnela i rozszczepienie chromatyczne (187)
  - o 7.4.1. Efekt Fresnela (187)
  - o 7.4.2. Rozszczepienie chromatyczne (188)
  - o 7.4.3. Parametry zależne od aplikacji (189)
  - o 7.4.4. Program wierzchołków (190)
  - o 7.4.5. Program fragmentów (191)
- 7.5. Ćwiczenia (193)

## **Rozdział 8. Mapowanie nierówności (195)**

- 8.1. Mapowanie nierówności ceglanej ściany (195)
  - o 8.1.1. Mapa normalnych ceglanej ściany (196)
  - o 8.1.2. Przechowywanie map nierówności jako map normalnych (197)
  - o 8.1.3. Proste mapowanie nierówności dla ceglanego muru (200)
  - o 8.1.4. Mapowanie nierówności dla rozbłysku (203)
  - o 8.1.5. Mapowanie nierówności na innej geometrii (206)
- 8.2. Mapowanie nierówności ceglanej podłogi (208)
  - o 8.2.1. Program wierzchołków dla renderingu obrazu ceglanej podłogi (210)
- 8.3. Mapowanie nierówności dla torusa (213)
  - o 8.3.1. Matematyka dotycząca torusa (213)
  - o 8.3.2. Program wierzchołków dla torusa z mapowaniem nierówności (216)
- 8.4. Mapowanie nierówności dla teksturowanych siatek wielokątnych (218)
  - o 8.4.1. Algorytm dla pojedynczego trójkąta (218)
  - o 8.4.2. Możliwe problemy (220)
  - o 8.4.3. Uogólnienie do siatek z wielokątów (222)
- 8.5. Połączenie mapowania nierówności z innymi efektami (223)
  - o 8.5.1. Standardowe tekstury (223)
  - o 8.5.2. Mapy połysku (223)
  - o 8.5.3. Rzucanie cieni na samego siebie (224)
- 8.6. Ćwiczenia (225)

## **Rozdział 9. Zagadnienia zaawansowane (227)**

- 9.1. Mgła (227)
  - o 9.1.1. Mgła jednorodna (228)
  - o 9.1.2. Atrybuty mgły (229)
  - o 9.1.3. Matematyka mgły (229)
  - o 9.1.4. Dostosowanie równań do zachowania zgodnego z intuicją (232)
  - o 9.1.5. Tworzenie jednorodnej mgły w programie Cg (233)
- 9.2. Rendering nierealistyczny (235)

- o 9.2.1. Cieniowanie jak w kreskówkach (235)
- o 9.2.2. Implementacja cieniowania kreskówkowego (236)
- o 9.2.3. Łączymy wszystko razem (239)
- o 9.2.4. Problemy związane z tym rozwiązaniem (241)
- 9.3. Rzutowanie tekstur (241)
  - o 9.3.1. W jaki sposób działa rzutowanie tekstur? (242)
  - o 9.3.2. Implementacja rzutowania tekstury (244)
  - o 9.3.3. Kod rzutowania tekstury (245)
- 9.4. Mapowanie cieni (248)
- 9.5. Łączenie (250)
  - o 9.5.1. Mapowanie pikseli z wejścia na wyjście (251)
  - o 9.5.2. Podstawowe operacje dotyczące łączenia (252)
- 9.6. Ćwiczenia (254)

## **Rozdział 10. Profile i wydajność (257)**

- 10.1. Opis profili (257)
  - o 10.1.1. Profil shadera wierzchołków dla DirectX 8 (257)
  - o 10.1.2. Podstawowy profil programu wierzchołków dla kart NVIDIA i OpenGL (258)
  - o 10.1.3. Profil programu wierzchołków ARB dla OpenGL (259)
  - o 10.1.4. Profil shadera wierzchołków dla DirectX 9 (259)
  - o 10.1.5. Zaawansowany profil programu wierzchołków dla kart NVIDIA i OpenGL (259)
  - o 10.1.6. Profile shadera pikseli dla DirectX 8 (260)
  - o 10.1.7. Podstawowy profil programu fragmentów NVIDIA dla OpenGL (261)
  - o 10.1.8. Profile shadera pikseli dla DirectX9 (261)
  - o 10.1.9. Profil programu fragmentów ARB dla OpenGL (262)
  - o 10.1.10. Zaawansowany profil programu fragmentów NVIDIA dla OpenGL (262)
- 10.2. Wydajność (263)
  - o 10.2.1. Korzystanie ze standardowej biblioteki Cg (263)
  - o 10.2.2. Zalety parametrów jednorodnych (264)
  - o 10.2.3. Program fragmentów a program wierzchołków (264)
  - o 10.2.4. Typy danych i ich wpływ na wydajność (265)
  - o 10.2.5. Wykorzystanie zalet wektoryzacji (265)
  - o 10.2.6. Kodowanie funkcji w teksturach (266)
  - o 10.2.7. Intensywnie wykorzystanie przemieszania i negacji (267)
  - o 10.2.8. Cienimy tylko te fragmenty, które musimy (267)
  - o 10.2.9. Krótszy kod assemblerowy nie zawsze jest szybszy (268)
- 10.3. Ćwiczenia (268)

## **Dodatek A Narzędzia Cg (269)**

- A.1. Pobieranie przykładów prezentowanych w niniejszej książce (269)
- A.2. Pobieranie narzędzia Cg Toolkit (269)

## **Dodatek B Biblioteka wykonywania Cg (271)**

- B.1. Czym jest biblioteka wykonywania Cg? (271)

- B.2. Dlaczego warto używać biblioteki wykonywania Cg? (271)
  - o B.2.1. Dostosowanie do nowszych procesorów graficznych (271)
  - o B.2.2. Brak problemów z zależnościami (272)
  - o B.2.3. Zarządzanie parametrami wejściowymi (272)
- B.3. W jaki sposób działa biblioteka wykonywania Cg? (273)
  - o B.3.1. Pliki nagłówkowe (274)
  - o B.3.2. Tworzenie kontekstu (274)
  - o B.3.3. Kompilacja programu (274)
  - o B.3.4. Wczytanie programu (275)
  - o B.3.5. Modyfikacja parametrów programu (276)
  - o B.3.6. Wykonanie programu (276)
  - o B.3.7. Zwalnianie zasobów (277)
  - o B.3.8. Obsługa błędów (277)
- B.4. Dodatkowe informacje (278)

## **Dodatek C Format pliku CgFX (279)**

- C.1. Czym jest CgFX? (279)
- C.2. Opis formatu (280)
  - o C.2.1. Techniki (280)
  - o C.2.2. Przebiegi (281)
  - o C.2.3. Stany renderingu (281)
  - o C.2.4. Zmienne i semantyka (282)
  - o C.2.5. Przypisy (282)
  - o C.2.6. Przykładowy plik CgFX (283)
- C.3. Moduły Cg obsługujące format CgFX (284)
- C.4. Dodatkowe informacje o CgFX (285)

## **Dodatek D Słowa kluczowe języka Cg (287)**

- D.1. Lista słów kluczowych języka Cg (287)

## **Dodatek E Funkcje standardowej biblioteki Cg (289)**

- E.1. Funkcje matematyczne (290)
- E.2. Funkcje geometryczne (293)
- E.3. Funkcje mapowania tekstur (294)
- E.4. Funkcje pochodnych (295)
- E.5. Funkcja testowania (296)

## **Skorowidz 297**