

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

Flash. Filmy i dźwięk. Techniki zaawansowane

Autor: praca zbiorowa

Tłumaczenie: Marek Binkowski

ISBN: 83-7197-854-5

Tytuł oryginału: [Rich Media StudioLab](#)

Format: B5, stron: 410



Pomyśl o dziesięciu najbardziej interesujących stronach internetowych, jakie widziałeś w ostatnim tygodniu. Co miały wspólnego? Niewiele, jeśli były naprawdę dobre. Oprócz tego, że przypuszczalnie wszystkie były utworzone we Flashu i niemal na pewno zawierały jakieś bogate elementy obrazu i dźwięku.

W tej książce chyba po raz pierwszy połączono te dwa aspekty – Flasha oraz bogate materiały filmowe i dźwiękowe. Pisarz Edward Bellamy trafnie przewidział wszechobecność mediów w roku 2000, obserwując Amerykę pod koniec 19. stulecia. Lecz nawet on myślał o bardziej jednolitym procesie. Podczas gdy większość świata zawierzyła hollywoodzkiej wizji filmu, czyli kinowych rozmiarów telewizorom stojącym w każdym mieszkaniu, bogate materiały filmowe zaczynają zaglądać na ekrany naszych komputerów połączonych z Internetem.

Prawdziwie bogaty film i dźwięk w projektach internetowych nie jest na razie rozpowszechniony, lecz kreatywność rozkwita wszędzie tam, gdzie pozwalają jej na to bariery technologiczne. Pomyśl o złotych latach filmu czy muzyki rozrywkowej. To, co dzieje się z dzisiejszymi mediami, może mieć większe znaczenie niż to, co udało się zrobić grupie The Beatles z czterościeżkowymi nagraniami.

To wzniosłe słowa, lecz poparcie dla nich możesz znaleźć, przeglądając chociażby pliki źródłowe na płycie CD-ROM, dołączonej do tej książki. W tym miejscu rozpoczyna się nowa podróż.

Rolą tego wstępu jest przywitanie cię na pokładzie, poinformowanie o konwencjach typograficznych stosowanych w książce, a także wskazanie, w którym miejscu możesz rozpocząć czytanie.

W pierwszym rozdziale, którego autorami są Murat Bodur i Doug McDermott, znajduje się opis jednej z wersji przyszłości bogatych mediów sieciowych. W rozdziale drugim znajdziesz przegląd opcji, jakie udostępnia Flash podczas pracy nad bogatymi materiałami filmowymi i dźwiękowymi. Wskażemy tam również, w których miejscach książki opisane są poszczególne opcje. Do dalszych rozdziałów raczej nie powinieneś przechodzić bez przeczytania poprzednich.

Wierzmy, że opisaliśmy wszystkie istotne opcje dotyczące opracowywania bogatych materiałów filmowych i dźwiękowych we Flashu i na stronach tej książki zawarliśmy wystarczająco dużo wiedzy nawet dla najbardziej dociekliwych umysłów.

Wydawnictwo Helion
ul. Chopina 6
44-100 Gliwice
tel. (32)230-98-63
e-mail: helion@helion.pl



Spis treści

Wstęp	13
Od czego zacząć	13
Konwencje typograficzne	14
Płyta CD-ROM	15
Wydawca oryginału	15
Rozdział 1. Rewolucja i ewolucja: spojrzenie na świat filmu i dźwięku.....	17
Internet	18
Materiały filmowe i dźwiękowe.....	19
Internetowe bankomaty	21
Przyszłość filmów flashowych.....	24
Rozdział 2. Planowanie projektu	27
Przygotowanie materiału wideo.....	27
Właściwy montaż	28
Prędkość odtwarzania klatek	28
Wymiary obrazu.....	29
Kompresja obrazu	30
Mamy materiał wideo, co dalej?.....	33
Importowanie materiału wideo do Flasha	33
Konwersja klipu wideo na format SWF.....	38
Umieszczanie zawartości flashowej w „zewnętrznych” środowiskach	40
Dźwięk	41
Plany i schematy	42
Rozdział 3. After Effects	47
Interfejs programu After Effects	47
Tworzenie efektów	51
Eksport materiału wideo dla Flasha.....	74
Tworzenie interfejsów	75
Przygotowanie materiałów w After Effects	76
Realizacja projektu we Flashu	79
Eksperymentowanie i tworzenie nowych efektów	81
Rozdział 4. Badcop: niska przepustowość, mocne wrażenia	83
Planowanie projektu Badcop	84
Metoda pracy	86
Kręcenie i przygotowanie materiału wideo dla Flasha.....	87
Importowanie projektu After Effects do Flasha	96
Optymalizacja	102

Dołączanie dźwięku	105
Końcowy rezultat	107
Rozdział 5. QuickTime i animacja	109
Materiał wideo.....	110
Wykorzystanie materiału wideo do tworzenia animacji	117
Rozdział 6. Mechanizm pobierania danych wideo we Flashu	137
Jak to działa.....	138
Plik FLA.....	139
Rozdział 7. Program Flix	145
Inne metody.....	150
Program Wildform Flix	151
Tworzenie banera w formacie SWF.....	154
Przygotowanie materiału wideo.....	155
A teraz coś z zupełnie innej beczki.....	164
Rozdział 8. Program Turbine Video Encoder	167
Jak to działa.....	167
Interfejs użytkownika	168
Otwieranie pliku źródłowego.....	170
Zaznaczanie fragmentu klipu	171
Kodowanie	173
Zakładka Video	174
Zakładka Flash	178
Zakładka Audio.....	182
Menu Settings i predefiniowane ustawienia	184
Przykładowy proces kodowania pliku Flasha.....	185
Rozdział 9. Tworzenie flashowej galerii wideo	191
Galeria wideo — co to takiego?	191
Tworzenie galerii wideo.....	193
Elementy sterowania	204
Suwak	210
Kod obsługujący suwak	214
Informacje o klipie wideo.....	219
Oprogramowanie pola informacyjnego	220
Czas na seans!	224
Rozdział 10. Flash, wideo i HTML	225
Znaczniki <OBJECT> i <EMBED>.....	226
Jak to naprawdę działa.....	228
Ramki	229
Przyciski w dokumencie HTML-a.....	231
Wreszcie trochę Flasha.....	233
Pójdźmy dalej.....	235
Inspiracja.....	236
Rozdział 11. Odtwarzacz QuickTime Flash	239
Planowanie	240
Tworzenie tła.....	241

Tworzenie masek	243
Planowanie materiału wideo w programie QuickTime 5 Pro	244
Klip wideo	245
Ścieżka dźwiękowa	248
Czołówka i informacje o autorach	250
Menu utworzone we Flashu	251
Dodatkowe przyciski sterujące	254
Kompilacja projektu w programie QuickTime 5 Pro	257
„Skóry”	262
Rozdział 12. Integracja odtwarzacza RealPlayer z Flashem.....	265
Wiodące odtwarzacze	266
Ogólne spojrzenie	267
Sterowanie odtwarzaczem RealPlayer za pomocą skryptów JavaScriptu.....	267
Obsługa błędów	267
Komunikacja z modulem rozszerzającym.....	268
Wysyłanie poleceń do odtwarzacza RealPlayer	269
Metoda DoPlay().....	269
Metoda DoStop().....	270
Metoda SetMute()	270
Metoda SetVolume()	270
Wskazywanie pliku do odtworzenia.....	270
Dalsze możliwości.....	271
Interfejs flashowy	272
Kompilacja elementów	274
Dołączanie zewnętrznego pliku ze skryptem JavaScriptu	274
Ustawianie globalnych właściwości za pomocą skryptu JavaScriptu	275
Osadzanie klipu wideo	275
Znaczniki <OBJECT> i <EMBED> osadzające film RealPlayer.....	275
Rozdział 13. Tworzenie dźwięku.....	277
Dlaczego dźwięk?	277
Wprowadzenie do dźwięku cyfrowego	279
Jakość dźwięku.....	281
Sprzęt	283
Dobór i łączenie sprzętu.....	283
Typowe konfiguracje.....	286
Oprogramowanie.....	295
Tworzenie prostej kompozycji	296
Moja wcześniejsza kompozycja	302
Jeszcze kilka porad i wskazówek	303
Nagrywanie materiału sesyjnego.....	307
Rozdział 14. Optymalizacja dźwięku.....	309
Optymalizacja dźwięku.....	309
Kompresja sygnału dźwiękowego	310
Korekcja widma	311
Program Sound Forge	311
Próbkowanie materiału sesyjnego.....	313
Moduły rozszerzające programu Sound Forge	317
Przetwarzanie ścieżki dźwiękowej.....	322

Tworzenie dźwięku na potrzeby Internetu	325
Dźwięk w formacie MP3	325
Ścieżki dźwiękowe	328
Tworzenie pętli i bridge'ów	331
Przygotowywanie pętli na potrzeby Flasha	337
Rozdział 15. Dynamiczne odtwarzanie dźwięku we Flashu.....	343
Możliwości dźwiękowe Flasha	343
Sterowanie dźwiękiem za pomocą skryptów	348
Mechanizm interaktywnej ścieżki dźwiękowej	354
Wybór i testowanie pętli	354
Interaktywna ścieżka dźwiękowa — przemyślenia	355
Tworzenie interaktywnej ścieżki dźwiękowej	356
Struktura mechanizmu	357
Integracja interaktywnej ścieżki dźwiękowej z innym plikiem FLA	363
Rozdział 16. Komercyjna prezentacja flashowa.....	369
Zadanie	369
Inspiracja	370
Rozwiązanie	371
Struktura projektu	374
Zanim uruchomisz Flasha	374
Składanie całości we Flashu	375
Dodatek A Teoria dźwięku cyfrowego.....	387
Natura dźwięku	388
Propagacyjna reprezentacja dźwięku	388
Sinusoidalna reprezentacja dźwięku	388
Prosta filtracja	390
Szeregi Fouriera i harmoniczne	390
Dźwięk cyfrowy	394
Szum.....	397
Skorowidz.....	407

Rozdział 14.

Optymalizacja dźwięku

W poprzednim rozdziale dowiedziałeś się, jak przygotować i połączyć sprzęt oraz oprogramowanie potrzebne do tworzenia materiału dźwiękowego. W tym rozdziale pokażemy, jak poddać utworzony materiał dźwiękowy dalszej obróbce, tak aby zoptymalizować go na potrzeby rozbudowanej prezentacji sieciowej.

Aplikacją, której użyjemy do cyfrowej obróbki materiałów sesyjnych utworzonych w poprzednim rozdziale, jest program Sound Forge. Istnieje również uboższa wersja tego programu, nosząca nazwę Sound Forge XP; demonstracyjną wersję programu można znaleźć pod adresem www.sonicfoundry.com.

Omawiany w tym rozdziale proces polega na optymalizacji dźwięku w celu zredukowania liczby danych i uzyskania pliku o rozmiarach pozwalających na używanie go w Internecie. Jednocześnie musi być zachowana najwyższa jakość dźwięku. Tak, to nie błąd — uzyskamy małe pliki z dźwiękiem o najwyższej jakości. Filtracja cyfrowa, której poddamy materiał dźwiękowy, jest bardzo podobna do tej stosowanej w komercyjnych produkcjach. Po opanowaniu tych technik będziesz potrafił wykonywać poszczególne etapy masteringu nagrań sesyjnych. Rezultatem tego procesu jest tak zwana „matka” — materiał najwyższej jakości, służący do tłoczenia płyt (zwykle pierwotnie zapisywany na taśmie DAT).

Zaraz na początku wyjaśnimy wszelkie wątpliwości dotyczące terminologii. W rozdziale będziemy używać terminu **kompresja** w sensie redukowania rozmiaru pliku i generowania małych plików na potrzeby Internetu, w szczególności w formacie MP3. Muzycy często używają tego samego terminu w innym kontekście — **kompresja** jest w muzyce efektem polegającym na wyrównaniu głośności poszczególnych partii utworu, dającym dźwięk o pełniejszym brzmieniu. Aby rozróżnić te dwa znaczenia (z obu będziemy korzystać), w przypadku drugiego będziemy się posługiwać terminem **kompresja sygnału**.

Optymalizacja dźwięku

Istnieją dwa rodzaje optymalizacji dźwięku — pierwszy polega na redukcji rozmiaru pliku; drugi prowadzi do poprawy jakości samego dźwięku. Większość projektantów sieciowych jest świadomych jedynie potrzeby optymalizacji rozmiaru pliku, jednak nie zdaje sobie sprawy z możliwości poprawiania jakości dźwięku poprzez odpowiednią obróbkę cyfrową. Tymczasem istnieją aż dwie metody takiej optymalizacji.

Po pierwsze, za pomocą odpowiedniej filtracji możesz uzyskać pełniejszy i bardziej zrównoważony dźwięk, bardziej „przyjazny” dla ucha słuchacza. Po drugie, możesz zoptymalizować sam przebieg dźwiękowy. Filtracja i kompresja audio mogą zmodyfikować dynamikę i kształt przebiegu dźwiękowego, co ułatwia późniejszą redukcję rozmiaru pliku bez wprowadzania słyszalnych szumów.

Podobnie jak w przypadku grafiki sieciowej, decydujące znaczenie ma jakość materiału wejściowego. Jeśli pierwotna ścieżka dźwiękowa jest niskiej jakości, wówczas finalny dźwięk przygotowany dla Internetu będzie co najmniej równie złej jakości. Proces próbkowania jest bardzo wrażliwy na szumy sygnału wejściowego, dlatego w przypadku zaszumionego źródłowego sygnału analogowego jego cyfrowy odpowiednik może ulec jeszcze większej degradacji. Podobnie szkodliwy jest zbyt niski poziom głośności pierwotnego sygnału.

Nagranie sesyjne z poprzedniego rozdziału (plik *ExampleSong_DATSMaster.wav*) zawiera dźwięk, który nie został poddany kompresji sygnału. Oznacza to, że występuje w nim duże zróżnicowanie głośności poszczególnych składników dźwięku. Ponadto istnieje duże zróżnicowanie głośności poszczególnych fragmentów utworu — rozpoczyna się on stonunkowo cicho, następnie staje się głośniejszy, choć w środkowej części ponownie pojawia się cicha partia. Utwór ten nagraliśmy, badając głośność tylko w najgłośniejszej jego części; chcieliśmy w ten sposób uniknąć przesycenia i zniekształceń sygnału, ponieważ tych zniekształceń nie sposób naprawić w dalszej obróbce sygnału. Oznacza to również, że dotychczas nie martwiliśmy się o jakość w cichych partiach utworu.

Kompresja sygnału dźwiękowego

W komercyjnych nagraniach kompresję sygnału dźwiękowego stosuje się po to, aby wysokie i niskie częstotliwości utworu były bardziej „przyjazne” dla słuchacza. Nie można dopuścić, by słuchacz przekręcał gałkę głośności, żeby wysłuchać cichych partii, po to by w dalszej części zostać zaskoczonym przez głośniejszą część utworu. Aby tego uniknąć, należy sprawić, by poziom głośności w skali całego utworu był mniej zróżnicowany. Do tego właśnie służy kompresja sygnału dźwiękowego. Posiada ona również działanie uboczne, polegające na tym, że może poprawić brzmienie głosu słabego wokalisty, którego głos po kompresji wydaje się lepiej kontrolowany.

Z punktu widzenia próbkowania zaletą skompresowanego sygnału jest mniej zmienny współczynnik szumu kwantyzacji, czyli szumu powstającego w procesie próbkowania (więcej informacji na ten temat znajdziesz w dodatku na końcu książki). Poziom tego szumu jest niezależny od poziomu sygnału, zatem szum kwantyzacji jest bardziej słyszalny w cichych partiach utworu. Zwiększając poziom sygnału w tych partiach, sprawiamy, że szum kwantyzacji jest lepiej zamaskowany. Poziom tego szumu pozostaje taki sam, lecz towarzyszą mu głośniejsze dźwięki (nawet w cichych partiach), dzięki czemu ludzkie ucho nie wychwytyuje tego szumu na całej długości utworu.

Korekcja widma

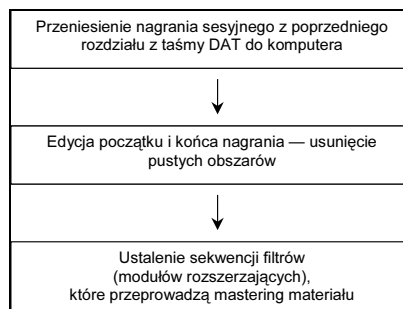
Korekcja widma (ang. *equalization*) również jest stosowana w komercyjnej produkcji dźwięku. Najczęściej służy ona do odfiltrowania skrajnie wysokich częstotliwości, które utrudniają słuchanie. Ponadto umożliwia wyeliminowanie skrajnie niskich dźwięków o częstotliwościach poniżej 50 Hz, które są potencjalnym zagrożeniem dla dużych systemów nagłośnieniowych. Utwór o skorygowanym widmie częstotliwości charakteryzuje się zrównoważonym rozkładem głośności niskich, średnich i wysokich tonów. Prowadzi również do zwiększenia jakości dźwięku i redukcji szumu kwantyzacji.

Skoro jesteśmy przy komercyjnej produkcji muzyki — większość utworów jest próbkowana z jakością płyty kompaktowej (rozdzielczość 16-bitowa i częstotliwość próbkowania 44,1 kHz); choć niektóre rodzaje muzyki są bardziej wymagające, jak na przykład muzyka klasyczna — w tym przypadku stosowana jest 32-bitowa rozdzielczość próbkowania, co pozwala uniknąć zniekształceń sygnału powstających przy niewystarczającej rozdzielczości próbkowania.

Aby przeprowadzić dalszą cyfrową obróbkę dźwięku, niektórzy profesjonaliści zwiększają rozdzielczość bitową do 32 bitów, szczególnie wówczas, gdy ma być zastosowanych szeregowo kilka filtrów. Jednak nie zastosujemy tego rozwiązania, ponieważ obecne oprogramowanie wewnętrznie i tak konwertuje dane na 32 bity w trakcie obliczeń, poza tym finalna płyta kompaktowa i tak zawiera dźwięk 16-bitowy. W naszym ćwiczeniu posługujemy się dźwiękiem o jakości płyty kompaktowej i nie zwracamy uwagi na przekłamania wynikające z filtracji cyfrowej. Jak wspomnieliśmy wcześniej, nie należy schodzić poniżej tej jakości, aż do momentu końcowej redukcji liczby danych przy konwersji pliku na potrzeby Internetu.

Proces przygotowania dźwięku na potrzeby prezentacji sieciowej, jaki tu przeprowadzimy, przedstawia rysunek 14.1.

Rysunek 14.1.



Program Sound Forge

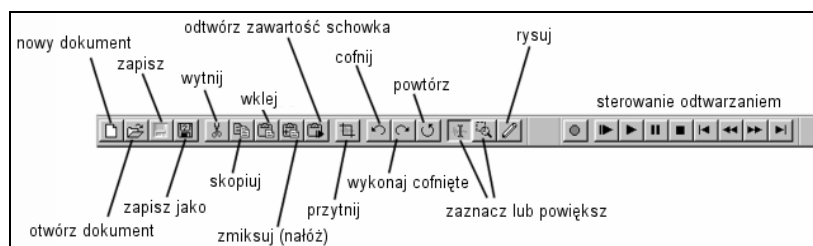
Zanim przystąpimy do obróbki dźwięku, spójrzmy na interfejs programu Sound Forge (rysunek 14.2). Wybraliśmy ten program, ponieważ jest najbardziej popularny i posiada możliwości porównywalne z konkurencyjnym oprogramowaniem. W odróżnieniu od aplikacji Cubase, program Sound Forge posiada bogatą dokumentację dostępną w formacie plików PDF, co może ci ułatwić głębsze jego poznanie.

Rysunek 14.2.



Jak widać na rysunku 14.2, większą część obszaru ekranu zajmują przebiegi dźwiękowe prawego i lewego kanału stereo. Po prawej stronie znajdują się wskaźniki poziomu sygnału, zaś po obu stronach poziomego paska przewijania znajdują się ikony sterujące powiększeniem obrazu. Za pomocą myszy możesz zaznaczać obszary przebiegów lub bezpośrednio edytować przebieg. Ikona w lewym górnym narożniku okna pozwala zdecydować, jaki ma być efekt zaznaczenia — ikona kursora pozwala zaznaczyć fragment przebiegu, ikona lupy umożliwi powiększenie wybranego fragmentu, natomiast ikona ołówka umożliwi narysowanie nowego kształtu przebiegu za pomocą myszy. Działanie większości elementów interfejsu jest stosunkowo intuicyjne (rysunek 14.3).

Rysunek 14.3.



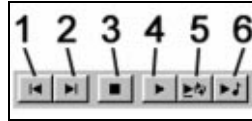
Oto kilka uwag na temat narzędzi przedstawionych na rysunku 14.3:

- ◆ Ikona „zmiksuj (nałoż)” umożliwia nałożenie zawartości schowka na istniejący przebieg dźwiękowy, począwszy od wskazanego miejsca. Umożliwia to selektywne miksowanie fragmentów utworu.
- ◆ Ikony „zaznacz lub powiększ” oraz „rysuj” działają tak samo jak wspomniana wcześniej ikona występująca w lewym górnym narożniku obrazu. Ikony te funkcjonują jako przełączniki (naraż jest aktywna tylko jedna z nich).
- ◆ Ikona ołówka („rysuj”) reprezentuje narzędzie, które umożliwia bezpośrednie rysowanie przebiegu dźwiękowego. Aby za jego pomocą modyfikować istniejący

przebieg, powiększ go maksymalnie, tak aby były widoczne poszczególne fale przebiegu. Za pomocą tego narzędzia można łatwo usuwać pojedyncze trzaski.

W lewym dolnym narożniku ekranu znajduje się podręczny zestaw ikon służących do sterowania odtwarzaniem (rysunek 14.4):

Rysunek 14.4.



Oto funkcje poszczególnych ikon:

1. Przejdź na początek przebiegu
2. Przejdź na koniec przebiegu
3. Zatrzymaj się i wróć do wskaźnika
4. Odtwarzaj w trybie normalnym
5. Odtwarzaj z zapętleniem
6. Odtwarzaj z uwzględnieniem zaawansowanych ustawień (funkcja nieużywana w tym rozdziale)

Próbkowanie materiału sesyjnego

W poprzednim rozdziale za pomocą sekwencera skomponowaliśmy utwór w formacie MIDI, który był (przypuszczalnie) odtwarzany przez kilka urządzeń MIDI. Sygnały dźwiękowe z tych oddzielnych urządzeń zmiksowaliśmy na pulpicie mikserskim i nagraliśmy na taśmie DAT. Teraz przeniesiemy tę ścieżkę dźwiękową z powrotem do komputera. Jeśli nie dysponujesz własną ścieżką dźwiękową, możesz skorzystać z mojego pliku sesyjnego — *ExampleSong_DATSMaster.wav*.

Oczywiście, jeśli korzystałeś z rozwiązania czysto programowego, takiego jak aplikacja Reason, w rezultacie otrzymałeś już cyfrowy plik dźwiękowy, zatem nie ma potrzeby jego ponownego próbkowania. Z kolei jeśli korzystasz z wewnętrznych kart muzycznych, możesz spróbować podłączyć ich wyjścia (liniowe lub optyczne) z wejściami głównej karty dźwiękowej i rejestrować dźwięk bezpośrednio podczas odtwarzania utworu. Takie rozwiązanie jest możliwe w przypadku niektórych konfiguracji programów, lecz jest zależne od wielu czynników, wymaga szybkiego komputera i nie można go stosować, korzystając z programu Cubase jako sekwencera... dlatego go nie polecamy.

Jeśli masz dostęp do drugiego komputera, możesz nim zastąpić magnetofon DAT i próbować dźwięk bezpośrednio za jego pomocą. W takim przypadku opisane niżej operacje powinieneś przeprowadzić na tym drugim komputerze. Dla profesjonalisty dużą korzyścią wynikającą z używania magnetofonu DAT jest fakt, że jest to ogólnie stosowany standard. Jeśli dalszą obróbką dźwięku mają się zająć inni fachowcy, będą bardzo zadowoleni, gdy dostarczysz im materiał na taśmie DAT.

Próbkowanie dźwięku

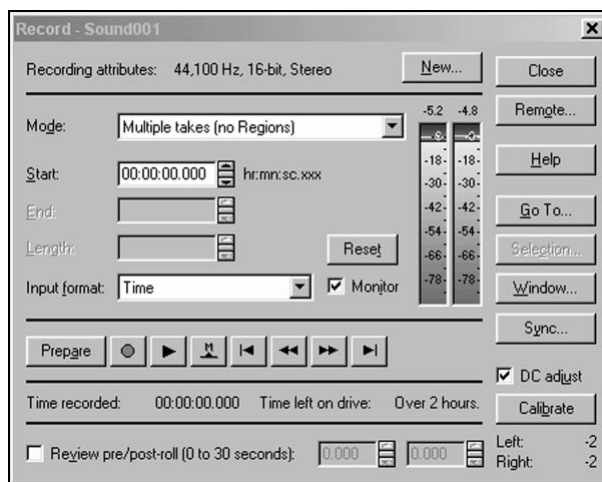
Jeśli w poprzednim rozdziale przygotowałeś własne nagranie sesyjne, użyj go w tym ćwiczeniu. Jeśli nie posiadasz żadnego własnego materiału, użyj mojego pliku sesyjnego — *ExampleSong_DATSMaster.wav*.

1. Ustaw mikser swej karty dźwiękowej tak, aby wyciszyć wszystkie wejścia oprócz *Line In* (wejścia liniowego) lub *SPDIF* (*Sony-Phillips Digital Interface* — wejście cyfrowe), do którego jest podłączone źródło rejestrowanego dźwięku. Aby otworzyć okno miksera systemowego, kliknij ikonę głośnika na pasku zadań Windows (użytkownicy komputera Macintosh powinni wybrać polecenie *Apple/Control Panels/Sound*).

Wyłączamy wszystkie inne wejścia (takie jak *FM Synthesis*, *CD Audio* itp.), nawet jeśli nie są do nich podłączone żadne urządzenia, ponieważ wejścia te mogą być źródłem szumu. Wnętrze komputera jest przepełnione szumem elektronicznym, pochodzącym od zasilacza, elektroniki działającej z wysoką częstotliwością, napędów elektrycznych, wirujących talerzy dysków twardych i urządzeń telekomunikacyjnych.

2. Wszelkie suwaki basów i sopranów, jakie mogą być dostępne dla twojej karty, ustaw w neutralnej (środkowej) pozycji — służą one wyłącznie do korygowania dźwięku samej karty i nie powinny brać udziału w przygotowywaniu dźwięku, który będzie odtwarzany na innym sprzęcie!
3. W programie Sound Forge kliknij ikonę *Record* (zapis). Pojawi się okno dialogowe *Record*, zawierające parametry związane z rejestracją i próbkowaniem dźwięku. Ustaw parametry tak, jak widać na rysunku 14.5.

Rysunek 14.5.



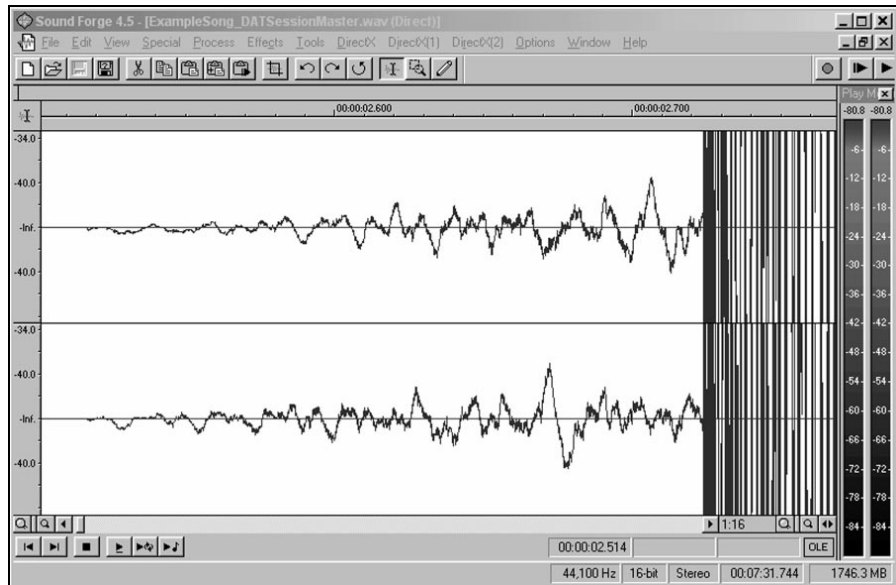
4. Uruchom odtwarzanie ścieżki dźwiękowej na zewnętrznym urządzeniu. Po zakończeniu nagrywania kliknij ikonę *Stop* (czarny kwadrat) w oknie *Record*. Na ekranie programu Sound Forge powinien się pojawić cały zarejestrowany przebieg (rysunek 14.6).

Rysunek 14.6.



5. Następnym krokiem jest usunięcie niepotrzebnych pustych obszarów, które bez wątpienia znajdują się na początku i na końcu zarejestrowanego przebiegu. Za pomocą paska przewijania i narzędzi powiększenia wyświetl na ekranie powiększony początkowy fragment przebiegu. W domyślnej skali może się wydawać, że początkowy fragment nie zawiera szumu, lecz gdy zwiększysz pionową skalę wyświetlania (za pomocą lupy po lewej stronie paska przewijania), przekonasz się, że jest tam szum (rysunek 14.7).

Rysunek 14.7.



6. Za pomocą myszy zaznacz pusty obszar poprzedzający utwór. Pozostaw jednak krótki fragment bezpośrednio poprzedzający utwór, aby uniknąć trzasku. Usuń zaznaczony fragment.

Oczywiście mógłbyś usunąć pusty obszar co do pojedynczego bajta, lecz nie polecam takiej praktyki. Sprzęt Hi-Fi, a także niektóre karty dźwiękowe są przystosowane do odtwarzania naturalnych dźwięków i mogą nie radzić sobie z nagłym wzrostem sygnału bez łagodnego przejścia. Tego typu nagły wzrost jest sztucznym zjawiskiem, możliwym do osiągnięcia tylko podczas edycji cyfrowej.

Zerowe wartości sygnału, przechodzące bezpośrednio w przebieg utworu, mogą również prowadzić do błędnego działania filtrów cyfrowych, zajmujących się wygładzaniem przebiegu dźwiękowego. Niektóre filtry korzystają z zaawansowanych algorytmów uśredniania, które w takim przypadku mogą dawać nieprawidłowe wartości na początku utworu. Może nie jest to długotrwałe zakłócenie, jednak pierwszych kilka sekund utworu to czas, kiedy jakość nagrania jest najbardziej słyszalna.

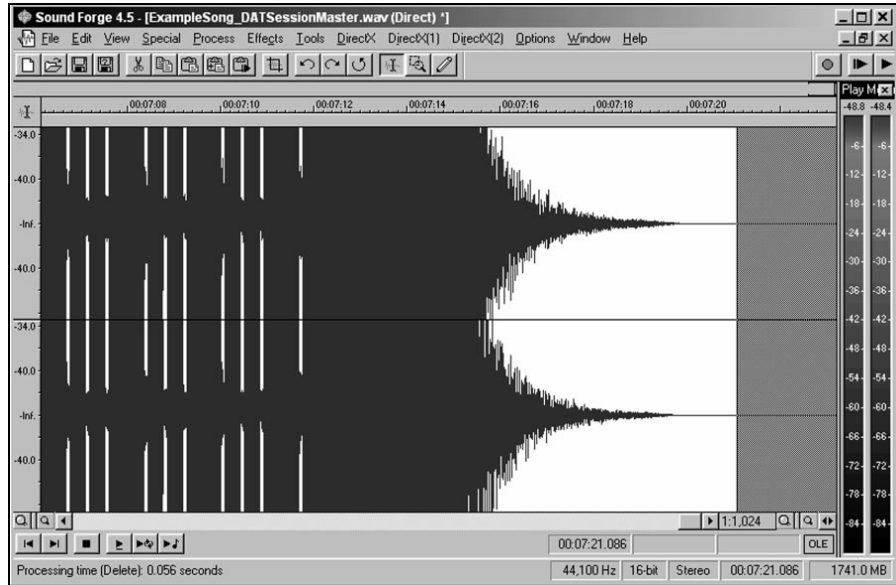
7. Aby uniknąć tych problemów, pozostawimy niewielki fragment ciszy na początku utworu, dodając efekt narastania (*Fade In*), aby „ostrzec” elektronikę przed rozpoczęciem odtwarzania utworu. Zaznacz pozostawiony fragment ciszy i wybierz polecenie *Process/Fade/In*. W zaznaczonym fragmencie program utworzy efekt narastania (rysunek 14.8).

Rysunek 14.8.



8. W podobny sposób zakończ utwór. Zaznacz ciszę występującą po jego zakończeniu i usuń ją, lecz pozostaw niewielki jej fragment. Następnie zaznacz pozostawiony fragment i wybierz polecenie *Process/Fade/Out*, aby uzyskać efekt wygasania (rysunek 14.9).

Rysunek 14.9.



Teraz mamy „czyste” nagranie sesyjne, które możemy poddać dalszemu przetwarzaniu. Jeżeli nie posiadasz programu Sound Forge, podobny efekt możesz uzyskać w innych aplikacjach. Na przykład w programie CoolEdit 2000 (<http://www.syntrillium.com/cooledit/>) zaznacz obszar, w którym chcesz uzyskać efekt narastania lub wygaszania, i wybierz polecenie *Transform/Amplitude/Amplify*. Następnie z listy predefiniowanych ustawień wybierz *Fade In* (w przypadku efektu narastania).

Moduły rozszerzające program Sound Forge

Przyjrzyjmy się teraz kilku modułom rozszerzającym program Sound Forge, służącym do obróbki dźwięku. Następnie zbudujemy łańcuch tych modułów, za pomocą którego przetworzymy naszą ścieżkę dźwiękową. Dzisiejsze komputery są wystarczająco szybkie i posiadają odpowiednie ilości pamięci, aby cyfrowo przetwarzać dźwięk. Oznacza to, że możesz kupić programowe wersje wielu sprzętowych efektów. Znani producenci sprzętowych przetworników dźwięku, tacy jak Lexicon, tworzą obecnie pełną gamę efektów zarówno w wersjach sprzętowych, jak i programowych, w postaci modułów rozszerzających. Istnieją również odwrotne sytuacje — programiści, którzy do tej pory tworzyli moduły rozszerzające do obróbki dźwięku, teraz tworzą również ich wersje sprzętowe!

W systemie Windows nowoczesne moduły rozszerzające, przetwarzające dźwięk, korzystają z technologii DirectX. Dlatego w magazynach muzycznych często można spotkać terminy „filtry DirectX”, „moduły DirectX”, czy „plug-iny DirectX”. Choć technologia ta jest związana przede wszystkim z grafiką trójwymiarową w grach komputerowych, można z niej korzystać we wszelkich zastosowaniach multimedialnych. Jedną z takich procedur nosi nazwę DirectShow i umożliwia przechwytywanie strumieni danych multimedialnych, do których należą również dane dźwiękowe. Programiści komputerów Macintosh dysponują podobnymi protokołami (VST, MAS, RTAS i Audio Suit). Większość komercyjnych modułów rozszerzających jest dostępnych w technologii DirectX oraz w większości protokołów macintoshowych.

Moduły rozszerzające DirectX, które przechwytyują dźwięk na poziomie DirectShow, nie są modułami konkretnego programu muzycznego, lecz samego systemu operacyjnego. Mogą one być używane przez dowolne oprogramowanie edytujące dźwięk, zgodne z technologią DirectX. Aby z nich korzystać, musisz mieć zainstalowaną aktualną wersję oprogramowania DirectX. Wówczas, po zainstalowaniu modułu, pojawia się on na liście modułów rozszerzających wszystkich zgodnych aplikacji dźwiękowych. Wiele z nich posiada wewnętrzne algorytmy operujące rozdzielczością bitową większą niż 16 bitów i, przetwarzając dźwięk, zachowują jego najwyższą jakość. Jest to ogromny krok naprzód w stosunku do tradycyjnych filtrów analogowych.

Dostępnych jest wiele modułów rozszerzających przetwarzających dźwięk, a także wiele innych efektów, których tu nie omówimy. Skorzystamy z jednego z najbardziej popularnych zestawów dla muzyki pop — łańcucha złożonego z modułu korekcji widma, kompresji sygnału dźwiękowego i maksymalizacji. Korzystamy z komercyjnych modułów, lecz w Internecie możesz znaleźć również ich darmowe odpowiedniki. Ustawienie parametrów dla każdego z modułów odbywa się w interfejsie zawierającym suwaki i inne elementy sterujące. Jeśli wiesz, jak ustawić analogową wersję filtra, jego cyfrowy odpowiednik zwykle jest ustawiany w taki sam sposób, więc nie powinieneś mieć z tym żadnych problemów.

Zanim użyjemy modułów rozszerzających do przetworzenia naszego nagrania, przyjrzymy się każdemu z nich i powiemy, w jaki sposób przetwarza on sygnał dźwiękowy.

Moduł korekcji widma

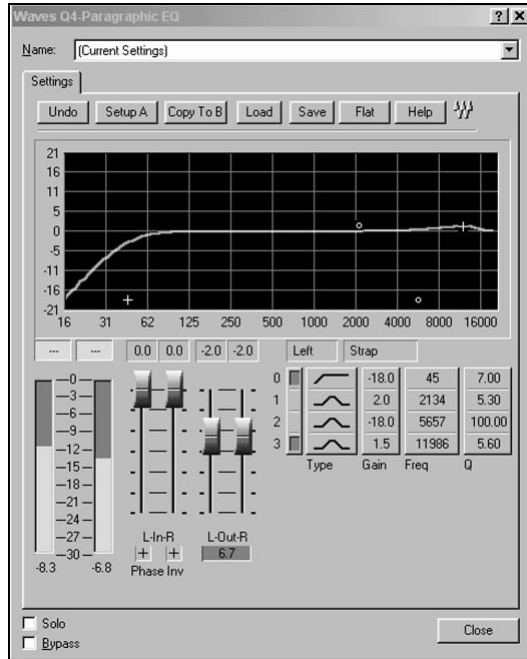
W naszym przykładzie użyjemy modułu korygującego widmo firmy Waves (www.waves.com). Aby skorzystać z tego modułu, po jego zainstalowaniu wybierz polecenie *DirectX/Waves Q4 Paragraphic EQ*. Jeśli dysponujesz modułem innego producenta, jego interfejs może się różnić od przedstawionego na rysunku 14.10. Najważniejszą sprawą związaną z ustawianiem tego filtra jest krzywa korekcji.

Oś pionowa wykresu reprezentuje wzmocnienie (w skali decybelowej — dB), zaś oś pozioma reprezentuje częstotliwości (Hz). Neutralnym ustawieniem, nie modyfikującym przebiegu dźwiękowego, jest pozioma linia na poziomie 0 dB. Gdy dany fragment krzywej jest wzniesiony ponad poziom 0 dB, odpowiednie częstotliwości zostaną wzmocnione. Na rysunku 14.10 widać, że w naszym przypadku niskie częstotliwości zostaną osłabione (krzywa poniżej poziomu 0 dB); ponadto częstotliwości w okolicach 10 kHz zostaną delikatnie wzmocnione. Nasz utwór zawiera dużą ilość basów, w związku z czym wysokie częstotliwości mogłyby zostać zbyt mocno skompresowane w następnym etapie.

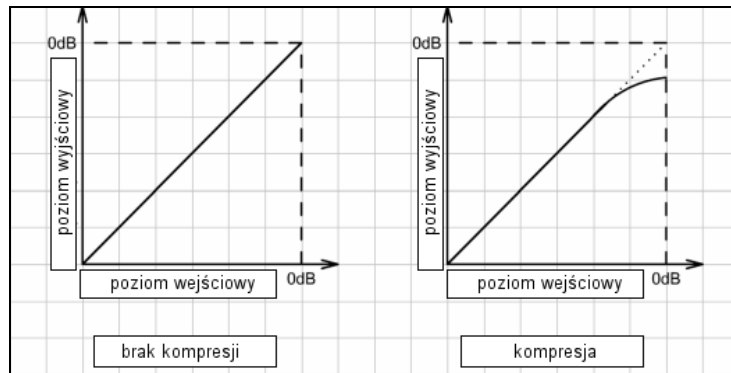
Moduł kompresji

Kompresja sygnału dźwiękowego modyfikuje poziomy głośności dźwięku, eliminując skrajnie wysokie i niskie poziomy głośności we wskazanym dźwięku lub obszarze kompozycji. Efekt ten daje bardziej zrównoważony przebieg dźwięku, dzięki czemu jest on bardziej „przyjazny” dla słuchacza. Wykresy na rysunku 14.11 przedstawiają zależność wyjściowego poziomu głośności sygnału od poziomu wejściowego. Wykres po lewej stronie reprezentuje brak kompresji.

Rysunek 14.10.



Rysunek 14.11.

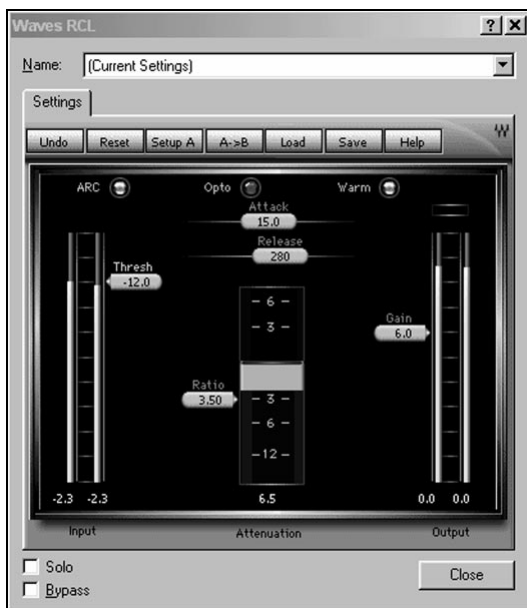


Ponieważ linia na tym wykresie jest prostą nachyloną pod kątem 45 stopni, wejście i wyjście filtra są takie same, zatem sygnał dźwiękowy nie jest kompresowany. Przerywane linie reprezentują punkty nasycenia (głośność w dB). Gdy sygnał wejściowy zbliża się do punktu nasycenia, tak samo zachowuje się sygnał wyjściowy. Na drugim wykresie widać odchylenie krzywej w górnej części, odpowiadającej głośnym dźwiękom. W tym przypadku, gdy sygnał wejściowy zbliża się do punktu nasycenia, sygnał wejściowy jest odsuwany do punktu nasycenia. Tak działa kompresja. Poziomy głośności powyżej punktu zakrzywienia są redukowane, dzięki czemu ogólna głośność dźwięku jest bardziej zrównoważona.

Działanie kompresora wykorzystującego wyłącznie taką krzywą byłoby zbyt drastyczne. Wyższej klasy kompresory posiadają różnego rodzaju funkcje uśredniające, zachowując pewną inercję poziomu kompresji, dzięki czemu wynikowy sygnał jest przyjemniejszy dla ucha.

Projektanci flashowi często wyposażają swoje menu i inne elementy interfejsu w procedury realizujące ruch inercyjny (z bezwładnością). Dzięki temu uzyskują efekt bardziej płynnego, naturalnego ruchu. W podobny sposób działa inercja poziomego kompresji — intensywność działania kompresora zmienia się z łagodną bezwładnością. Używany przez nas efekt, Waves RCL (ang. *Renaissance Compressor Limit*), emuluje filtr z taką inercją, dzięki czemu efekt kompresji jest bardziej zrównoważony (polecenie *DirectX/Waves RCL*) (rysunek 14.12).

Rysunek 14.12.



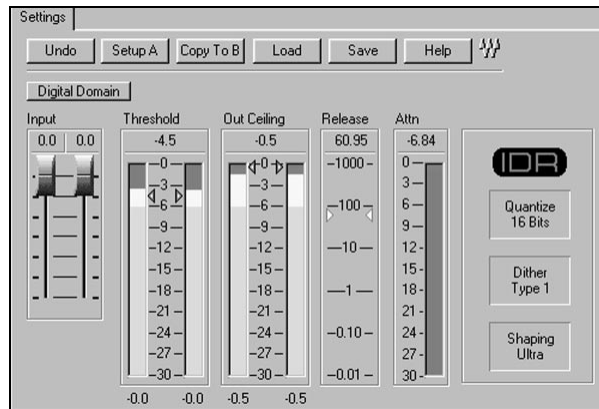
Ustawianie parametrów kompresora sygnału nie mieści się w zakresie tematycznym tej książki. Tak czy inaczej, jest to bardziej sprawa doświadczenia i praktyki, niż sztywnych reguł. Pokazane tu ustawienie może być dla ciebie dobrym punktem startowym do dalszego eksperymentowania. Korzystając z gotowych ustawień kompresora w programie CoolEdit, możesz szybko opanować tajniki działania tego typu przetworników. Wczytaj do tego programu nagranie sesyjne, wybierz gotowy schemat kompresji sygnału i poeksperymentuj z krzywą kompresji, dodając i przeciągając punkty kontrolne.

Moduł maksymalizacji

Połączenie korekcji widma i kompresji było typowym procesem masteringu muzyki pop aż do roku 1990. Wówczas wynaleziono nowy typ kompresji, noszący nazwę maksymalizacji. Najpierw pojawił się w formie sprzętowej, która była bardzo droga. Kilka lat później pojawiły się tańsze wersje programowe tego efektu. Dziś są używane chyba przez wszystkich i są odpowiedzialne za powstanie nowych nurtów muzycznych wykorzystujących ciężkie basowe dźwięki. Technologia pozwalająca na osiągnięcie tego typu efektu nie była dostępna przed rokiem 1990, więc każdy posiadacz kolekcji płyt pochodzących sprzed tego okresu zauważy, że starsze oryginalne nagrania wydają się dziś „ciche”.

Ostatni moduł w naszym łańcuchu filtrów nosi nazwę Wave Maximizer (rysunek 14.13). Jego działanie polega na dynamicznym zwiększaniu poziomu sygnału w kierunku punktu nasycenia, jednak bez osiągnięcia go. W efekcie wszystkie komponenty utworu uzyskują lepszy poziom nagrania. Proces ten różni się jednak od globalnego zwiększania głośności dźwięku, ponieważ moduł Maximizer opiera swoje działanie na charakterystyce częstotliwościowej, a nie prostej amplitudzie sygnału. Dzięki temu dźwięki ciche, harmoniczne stają się głośniejsze, zaś głośne zostają delikatnie wyciszone; poszczególne instrumenty są wyraźniejsze i lepiej zdefiniowane, przy czym zwiększa się ogólna energia sygnału dźwiękowego. Efekt ten może mieć większy wpływ na jakość dźwięku, niż można by się tego spodziewać.

Rysunek 14.13.



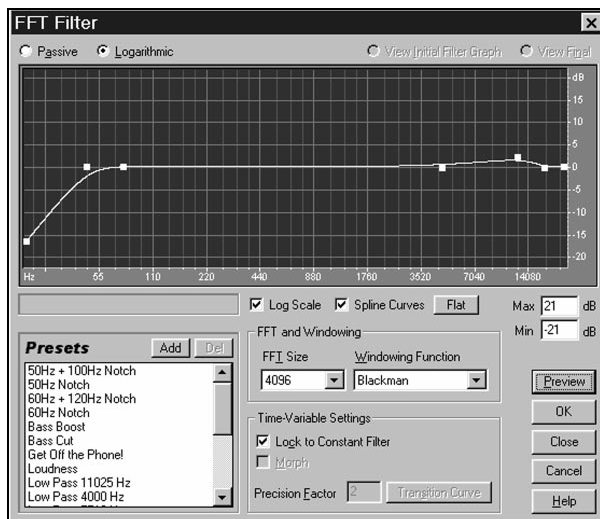
Filtry maksymalizacji mogą być drogie, lecz mają one zbawienny wpływ na dźwięki przygotowywane dla Internetu, z dwóch powodów. Po pierwsze zmaksymalizowany sygnał zawiera mniej komponentów o niskim poziomie głośności, zatem wszelki szum jest lepiej maskowany. Dotyczy to zarówno obecnie wykonywanych operacji, jak i przyszłych efektów i zmian, jakie możesz wprowadzić, w szczególności polegających na zmniejszeniu rozdzielczości bitowej przy przygotowywaniu pliku na potrzeby Internetu. Dzięki temu dźwięk może się mieścić w mniejszym pliku, zachowując stosunkowo wysoką jakość.

Po drugie, schemat kodowania MP3 szczególnie „lubi” zmaksymalizowany dźwięk — kodowanie w tym formacie jest znacznie wydajniejsze w przypadku przetworzonego w ten sposób dźwięku. Wynika to z faktu, że większa część wysokich częstotliwości jest skutecznie maskowana przez wzmocnione niskie, basowe częstotliwości. Czasem umożliwia to nawet wybranie niższej gęstości strumienia bitowego przy zachowaniu tej samej pozornej jakości dźwięku. Użyliśmy słowa „pozornej”, bo w tym przypadku sam przebieg dźwięku ulega modyfikacji, jednak ludzkie ucho jej nie wychwytuje ze względu na wzmocnione inne częstotliwości. A liczy się właśnie to, co słyszymy (nie wierz zatem wszystkim, o czym usłyszysz!).

Pierwsze dwa filtry są dostępne dla wielu shareware’owych programów, takich jak CoolEdit. Filtry kompresji znajdziesz, wybierając polecenie *Transform/Amplitude/Dynamic Processing* (dostępnych jest tak kilka gotowych ustawień kompresji, które możesz modyfikować na bieżąco, sprawdzając wynik za pomocą przycisku *Preview*). Aby w programie CoolEdit przeprowadzić korekcję widma, możesz kupić odpowiedni moduł rozszerzający lub utworzyć filtr FFT (ang. *Fast Fourier Transform* — szybka transformata

Fouriera). W tym celu wybierz polecenie *Transform/Filters/FFT Filter*, a następnie narysuj krzywą korekcji w oknie *FFT Filter* (rysunek 14.14). Aby zmodyfikować kształt krzywej, kliknij krzywą, dodając tym samym punkt kontrolny, który możesz następnie przeciągnąć. Jeśli chcesz dokładnie umiejscowić punkt dla częstotliwości 0 Hz, zaznacz opcje *Log Scale* i *Spline Curves* oraz przypisz parametrom *Min* i *Max* wartości odpowiednio 21 dB i -21 dB.

Rysunek 14.14.



Przetwarzanie ścieżki dźwiękowej

Poznałeś już moduły, za pomocą których przetworzymy naszą ścieżkę dźwiękową. Utworzymy teraz łańcuch tych modułów. Rysunek 14.15 przedstawia ich kolejność w łańcuchu. Ze struktury tej wynika, że rezultat przetwarzania otrzymywany na wyjściu pierwszego modułu jest podawany na wejście następnego itd.

Rysunek 14.15.

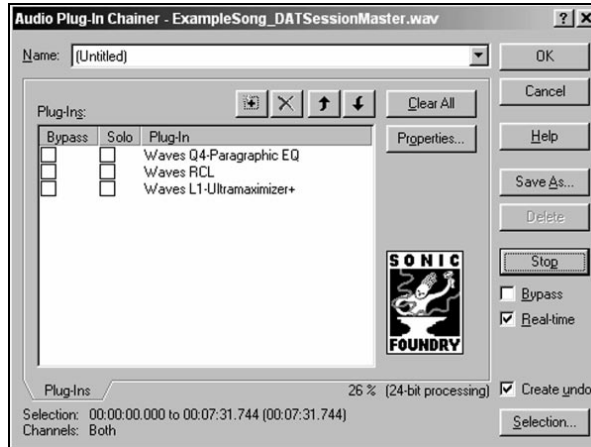


Gdy już ustawisz parametry poszczególnych modułów w ich oknach dialogowych (użyj ustawień pokazanych na rysunkach, jeśli nie jesteś pewien ich działania lub chcesz uzyskać dokładnie taki rezultat, jak w książce), możesz przystąpić do tworzenia łańcucha modułów.

W programie Sound Forge wybierz polecenie *Tools/Audio plugin chain*.

Pojawi się okno dialogowe *Audio Plug-In Chainer* (rysunek 14.16) (menu udostępniające moduły DirectX pojawi się tylko w przypadku, gdy program wykryje w systemie zainstalowany przynajmniej jeden moduł rozszerzający).

Rysunek 14.16.



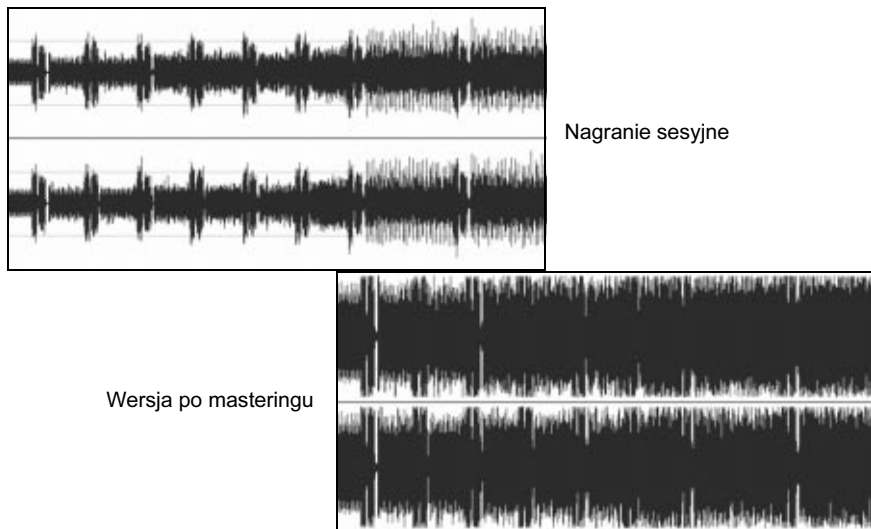
Umieszczanie modułów w łańcuchu jest bardzo podobne do umieszczania akcji w oknie *Actions* Flasha, w trybie *Normal Mode*. Oto, jak działają poszczególne ikony:

- ♦ Ikona ze znakiem + (plus) umożliwia dodanie do łańcucha nowego modułu.
- ♦ Ikona ze znakiem × (krzyżyk) umożliwia usunięcie wskazanego modułu z łańcucha.
- ♦ Ikony ze znakami strzałek w górę i w dół umożliwiają przesuwanie modułu w kierunku początku lub końca łańcucha.

Gdy utworzysz łańcuch modułów, kliknij przycisk *OK*.

Wynikową ścieżkę dźwiękową, uzyskaną w wyniku przetwarzania nagrania sesyjnego zapisaliśmy w pliku *ExampleSong_ProdMaster.wav*. Posłuchaj, jak bardzo znasterowana ścieżka różni się od nagrania sesyjnego (zapisanego w pliku *ExampleSong_DATSMaster.wav*). Rysunek 14.17 przedstawia jego sesyjną wersję nagrania i wersję po masteringu. Widać wyraźną różnicę!

Rysunek 14.17.



Na pierwszy rzut oka może się wydawać, że wersja zmastrowana jest przesycona i zniekształcona, jednak w rzeczywistości jej brzmienie jest czystsze i wyraźniejsze. Nie mają miejsca żadne przesylenia, ponieważ nie pozwoliłby na to moduł rozszerzający. Zmasterowana wersja niemal przez cały czas zachowuje ten sam, wysoki poziom sygnału, czyli korzysta z pełnej skali wartości, co minimalizuje słyszalny szum kwantyzacji. W porównaniu do niej amplituda wersji sesyjnej jest mocno zmienna. Amplituda fali odpowiada odbieranej głośności sygnału, dlatego obserwując przebiegi, możesz odnieść wrażenie, że mastering zakłócił dynamikę kompozycji — ciche fragmenty mają niemal taką samą widzialną amplitudę, jak głośne.

Stałoby się tak w sytuacji, gdyby ktoś operował pokrętkiem głośności na wzmacniaczu, starając się zachować cały czas taką samą słyszalną głośność utworu, niwelując tym sposobem wszelkie zmiany w jego dynamice. Jednak w naszym przypadku nic takiego się nie dzieje — ludzkie ucho nadal odbiera ciche fragmenty jako ciche, a głośne jako głośne; różnice głośności pomiędzy nimi są zachowane, choć nie jest to widoczne na „gęstym” wykresie amplitudy.

Nasz łańcuch modułów przetwarzających dźwięk utworzył znormalizowaną wersję pliku dźwiękowego (w normalizacji brały udział dwa pierwsze moduły), a następnie zmaksymalizowaną za pomocą trzeciego modułu. Może się wydawać, że działanie pierwszych dwóch modułów nie jest aż tak istotne, jednak pamiętaj, że trzeci moduł wydatnie wzmacnia sygnał; jeśli nie byłby on uprzednio znormalizowany, mogłyby powstać efekty zbytniego wzmocnienia.

Czego nie należy robić

Wiele osób tworzących własne ścieżki dźwiękowe wpada w pułapkę, polegającą na używaniu regulatorów barwy dźwięku we wzmacniaczu podczas tworzenia i edycji dźwięku. Wierzą oni, że finalna ścieżka dźwiękowa ma zrównoważone basy i soprany, lecz gdy zostanie ona odtworzona na zwykłych głośnikach komputerowych, poza studiem nagraniowym, okazuje się, że brzmi płasko. Aby tego uniknąć, nie odtwarzaj komponowanej ścieżki na sprzęcie z podbitym basem lub żadną inną korekcją barwy. Podczas edycji słuchaj surowego sygnału i modyfikuj go, aż będzie brzmiał dobrze w takiej konfiguracji.

Skrajnie niskie dźwięki basowe i częstotliwości poniżej 50 Hz mogą brzmieć fantastycznie w głośnikach o mocy 30 W w twojej sypialni, lecz mogą one „usmażyć” duże systemy nagłaśniające, dlatego zawsze je odfiltruj. Domowy sprzęt Hi-Fi może z łatwością zostać uszkodzony przez nietypowe dźwięki, jakie można uzyskać w procesie obróbki nagrań. Choć to kuszące, nie odtwarzaj ścieżki z dużą głośnością, zanim nie poddasz jej kompresji. (Przy okazji: dowolny głośnik, który nie został zaprojektowany do bezpośredniego wzmacniania dźwięku gitary, może się spalić nawet przy zaskakująco niskim poziomie głośności, jeśli podłączysz gitarę bezpośrednio do wejścia liniowego lub mikrofonowego; dlatego nie rób takich eksperymentów, jeśli nie jesteś pewien, że wiesz wszystko o silnych impulsach elektrycznych, jakie mogą generować gitary elektryczne).

Tworzenie dźwięku na potrzeby Internetu

Utworzyliśmy plik dźwiękowy o jakości odpowiadającej standardom komercyjnym. Przeprowadzony mastering sprawił, że uzyskaliśmy dźwięk nie tylko o lepszym brzmieniu, lecz również z mniej słyszalnym szumem kwantyzacji i optymalnym poziomem głośności. Jest to znakomity materiał do utworzenia dźwięku na potrzeby Internetu. Istnieje kilka formatów, do których możemy skonwertować naszą ścieżkę dźwiękową. Oto trzy najbardziej użyteczne:

- ♦ Samodzielny plik MP3 przeznaczony do pobrania i odtworzenia na lokalnym komputerze
- ♦ Samodzielny plik MP3 przeznaczony do odtwarzania strumieniowego
- ♦ Pętle dźwiękowe przeznaczone do umieszczenia w prezentacjach i aplikacjach multimedialnych

W pozostałej części tego rozdziału zajmiemy się dwoma pierwszymi punktami. W następnym rozdziale pokażemy zaś, jak pętle utworzone na bazie wysokiej jakości materiału dźwiękowego można wykorzystać w gotowym, lecz bezdźwięcznym projekcie flashowym.

Dźwięk w formacie MP3

Jednym z ważniejszych zastosowań Internetu dla przeciętnego użytkownika jest pobieranie muzyki. Pomimo faktu, że nielegalna dystrybucja muzyki w formacie MP3 jest nieustannie piętnowana przez posiadaczy praw autorskich, legalne pliki MP3 są dobrą metodą dystrybucji muzyki w Internecie. Dalszy los tej muzyki jest różny — użytkownicy konwertują pliki MP3 na nieskompresowane pliki dźwiękowe i wypalają je na płytach CD-R, umieszczają je w osobistych odtwarzaczach MP3 lub po prostu odtwarzają na swoich komputerach. Jak wspomnieliśmy wcześniej, z czasem zwiększy się zapotrzebowanie różnego rodzaju zespołów muzycznych na strony Internetowe, za pomocą których będzie rozpowszechniana ich twórczość. Choćby z tego względu powinniśmy wiedzieć, jak tworzyć dobrej jakości pliki MP3.

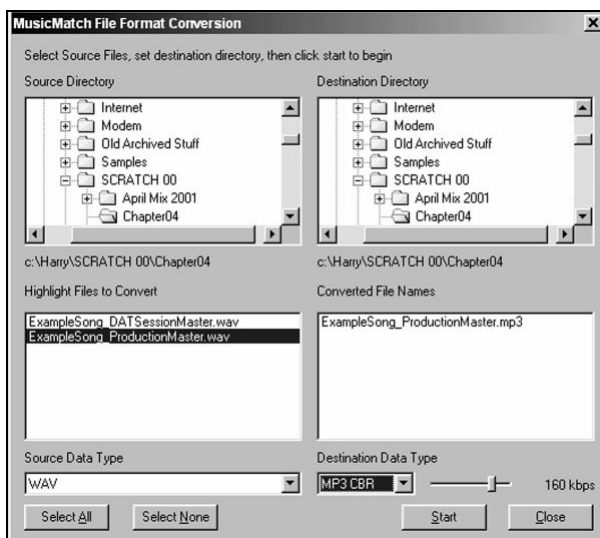
Istnieje wiele prostych konwerterów dźwięku na format MP3, przy czym w Internecie można znaleźć sporą grupę darmowego oprogramowania tego typu. Zdecydowaliśmy się wybrać program MusicMatch (www.musicmatch.com). Jest on nie tylko jednym z najłatwiejszych w obsłudze, lecz również jest dostępny dla systemów Windows, Mac OS i Linux (rysunek 14.18). Występuje w podstawowej wersji „Basic” oraz wzbogaconej „Plus”; wersja „Basic” jest wystarczająca dla naszych zastosowań i jest darmowa! Co najlepsze, Flash potrafi importować pliki MP3 tworzone przez ten program. Flash, podobnie jak kilka innych programów, jest bardzo kapryśny, jeżeli chodzi o importowanie plików MP3 — nie akceptuje on plików generowanych przez kilka innych konwerterów.

Rysunek 14.18.



Proces tworzenia pliku MP3 wygląda w programie MusicMatch podobnie, jak w innych konwerterach; aby go rozpocząć, wybierz polecenie *File/Convert*. Pojawi się okno dialogowe *File Format Conversion* (rysunek 14.19). W górnych dwóch przewijanych polach możesz wybrać katalog źródłowy (*Source Directory*) i docelowy (*Destination Directory*). W dolnych polach wyświetlane są nazwy plików źródłowych i wynikowych. W polu *Destination Data Type* możesz wybrać typ formatu MP3 oraz docelową gęstość strumienia bitowego. Zobaczmy, jakie opcje udostępnia to pole.

Rysunek 14.19.



Opcja MP3 CBR

Opcja *MP3 CBR* umożliwia generowanie plików MP3 o stałej gęstości strumienia bitowego (*CBR* — *Constant Bit Rate*). Z tego względu jest przydatna w przypadku strumieniowego odtwarzania dźwięku — nie musisz martwić się o końcowy rozmiar pliku, lecz raczej o wymagania dotyczące przepustowości łącza i dostępności. Zwykle przed rozpoczęciem konwersji możesz sprawdzić, jaką część strumienia bitowego możesz przeznaczyć na dźwięk w swojej aplikacji i wówczas możesz ustawić odpowiednią wartość parametru.

Jeśli prezentację kierujesz do użytkowników łączy od dużej przepustowości, możesz ustawić gęstość strumienia na 80 kilobitów na sekundę. Jeszcze wyższych wartości możesz użyć w przypadku, gdy plik ma być pobierany w całości i odtwarzany na lokalnych komputerach użytkowników. Jednak w tym przypadku tak czy inaczej sugerowałbym skorzystanie z opcji *MP3 VBR* (zmiennej gęstości strumienia bitowego). Nieskompre-

sowany plik dźwiękowy w formacie WAV o rozmiarze 24 MB i jakości płyty kompaktowej można skonwertować na plik MP3 o gęstości strumienia 80 kb/s i wielkości 1,3 MB. We Flashu zwykle są używane niższe gęstości strumienia bitowego dla dźwięku (Flash eksportuje dźwięk w formacie MP3 ze stałą gęstością strumienia bitowego). W przypadku łączu o niskiej przepustowości (modem 56 kb/s) używane są wartości poniżej 40 kb/s.

Eksperymenty z naszym nagraniem sesyjnym i jego zmasterowaną wersją ujawniły, że w wyniku optymalizacji wersja po masteringu, skonwertowana do pliku MP3 o gęstości strumienia bitowego 64 kb/s, brzmi równie dobrze, jak nagranie sesyjne skonwertowane do pliku MP3 o gęstości strumienia 80 kb/s. Sprawdziliśmy również wykres widmowy obu wersji i okazało się, że choć wersja po masteringu i konwersji ma niższą częstotliwość maksymalną, amplitudy jej wysokich częstotliwości ogólnie są wyższe. Przypuszczalnie to jest przyczyną wrażenia, że utwór zawiera więcej wysokich częstotliwości i w konsekwencji wyższą jakość.

Opcja MP3 VBR

Opcja *MP3 VBR* generuje pliki MP3 o zmiennej gęstości strumienia bitowego (*VBR* — *Variable Bit Rate*). W związku z tym trudno ustalić wymagania dotyczące przepustowości łącza. Podczas odtwarzania pliku ilość przesyłanych danych bezustannie się zmienia, zatem format ten jest mniej użyteczny dla takiego sposobu odtwarzania. Z drugiej strony format taki jest bardziej wydajny z punktu widzenia jakości dźwięku i stopnia kompresji — większa ilość danych przypada na bardziej wymagające fragmenty utworu, zaś w pozostałych partiach dźwięk jest opisywany przez mniejszą ilość danych. W związku z tym warto stosować ten format w przypadku plików MP3, które mają być w całości pobrane i odtwarzane na komputerze lokalnym.

Niektóre aplikacje przechwytyją wszystkie pliki MP3, które próbujesz pobrać, i automatycznie rozpoczynają ich odtwarzanie. Zwykle nie taki jest zamiar użytkownika łącza o szybkości 56 kb/s, który chce pobrać plik MP3 o rozmiarze 5 MB. Ponadto wielu posiadaczy praw autorskich posiada programy poszukujące plików MP3 udostępnionych do pobrania i stosują oni dosyć obcesowe założenie, że wszystkie pliki MP3 łamią prawa autorskie. Aby uniknąć kłopotów z tym związanych, warto umieszczać pliki MP3 przeznaczone do pobrania w archiwach ZIP lub zmieniać ich nazwę, tak aby nie miały rozszerzenia MP3 (informując o tym finalnego użytkownika).

Format MP3 VBR pozwala skompresować oryginalny plik nawet dwudziestokrotnie z zachowaniem dobrej jakości dźwięku, jednak znacznie lepsze rezultaty uzyskasz, jeśli przed konwersją poddasz dźwięk masteringowi. Nasza zmasterowana ścieżka dźwiękowa zachowała dobrą jakość nawet wówczas, gdy zmieściliśmy ją w pliku o rozmiarze stanowiącym zaledwie kilka procent rozmiaru oryginalnego pliku.

W tym przypadku prawie nie słyszeliśmy różnicy, jednak porównaliśmy wykresy widmowe obu plików. Skonwertowana wersja utraciła wszystkie częstotliwości wyższe niż 15 kHz. Te częstotliwości nie są odbierane przez wiele osób (szczególnie starszych), jednak ich brak może również wpłynąć na gorszą jakość niższych częstotliwości, wynikającą z szumu powstającego przy obniżeniu częstotliwości próbkowania. W naszym przypadku problem ten jest prawie niesłyszalny, ponieważ wysoki poziom głośności

poszczególnych częstotliwości maskuje szum kwantyzacji. Reasumując, dzięki masteringowi uzyskaliśmy słyszalnie dobrą jakość dźwięku nawet po kilkudziesięciokrotnej redukcji rozmiaru pliku.

Opcja WMA

Format *Windows Media Audio* (WMA) jest jedynym formatem odtwarzanym przez niektóre przenośne odtwarzacze. Firmy nagraniowe uwielbiają ten format, ponieważ umożliwia on wprowadzanie ograniczeń czasowych związanych z prawami autorskimi, zatem pobrane pliki po upływie określonego czasu stają się bezużyteczne. Format WMA jest porównywalny z formatem MP3 CBR ze względu na stopień kompresji, jednak zalecamy wybór formatu MP3, ponieważ jest on bardziej popularny i powszechniej obsługiwany (chyba że przygotowujesz pliki specjalnie dla odtwarzaczy przenośnych WMA — wówczas wybór jest prosty).

Opcja RealAudio

Kolejnym pretendentem jest format RealAudio, lecz nadal jest mniej popularny od formatu MP3, poza tym nawet dzisiaj nie jest obsługiwany przez niektóre serwery. Jeśli nie masz specjalnego powodu, by używać formatu RealAudio (na przykład uporu klienta), zalecamy użycie formatu MP3.

Konkluzja

Konwertując plik WAV do formatu MP3, przekonaliśmy się, że użycie wysokiej jakości materiału dźwiękowego po masteringu pozwala zminimalizować rozmiar pliku. Wynika to z faktu, że dzięki wysokiemu poziomowi głośności poszczególnych komponentów ścieżki dźwiękowej, można lepiej ukryć wszelkie zniekształcenia i artefakty i tym samym opisać falę dźwiękową za pomocą mniejszej liczby danych. Choć rzeczywisty kształt fali ulegnie zniekształceniu, dzięki niedoskonałości ludzkiego ucha ta utrata jakości pozostanie niezauważona. Nasza ciężka praca była owocna — uzyskaliśmy niewielki plik dźwiękowy o wysokiej jakości.

Ścieżki dźwiękowe

W poprzednim podrozdziale pokazaliśmy, jak generować pliki dźwiękowe, które mogą być samodzielnie odtwarzane i rozprowadzane w Internecie. Jednak co nas bardziej interesuje, to dźwięk odtwarzany wraz z bogatym obrazem, grafiką animowaną lub filmem — tego typu prezentacje tworzymy przecież w programach Flash czy Director. Dźwięk jest też używany w prostszej formie na istniejących witrynach — wzbogaca on kliknięcia przycisków i inne operacje związane z interfejsem (choć nie można w tym przypadku mówić o „bogatych” multimedialach). W tym podrozdziale zajmiemy się specyficznymi cechami ścieżek dźwiękowych dołączanych do prezentacji multimedialnych. Pokażemy, co należy zrobić, aby ścieżka dźwiękowa była interaktywna. Będziemy teraz pracować w innym środowisku, niż do tej pory, stawiającym bardziej złożone wymagania.

W przypadku prezentacji multimedialnych strumień danych dźwiękowych musi współistnieć ze strumieniem danych graficznych, opisujących zawartość obrazu. Dźwięk może wpłynąć na znaczne zwiększenie rozmiaru plików Flasha czy Directora, szczególnie w przypadku braku optymalizacji. W przypadku grafiki możesz poczynić duże oszczędności danych, korzystając wielokrotnie z tych samych symboli i elementów graficznych. Spróbujemy w ten sam sposób postąpić z dźwiękiem. W tym celu podzielimy ścieżkę dźwiękową na użyteczne fragmenty.

Ważny jest tu jeszcze jeden aspekt. Elementy grafiki są wielokrotnie wykorzystywane, lecz są one również interaktywne — obraz zmienia się w zależności od działań użytkownika. Być może ścieżka dźwiękowa również powinna być interaktywna. Może na przykład zmieniać się w zależności od tego, co uczyni użytkownik. Wymaga to sporych przemyśleń na temat tworzenia poszczególnych fragmentów ścieżki dźwiękowej i nimi się teraz zajmujemy. Projekt utworzymy we Flashu.

Zasada tworzenia dźwięku, który można wielokrotnie wykorzystać, jest prosta — tworzymy tak zwane pętle. Pętla to niewielki fragment utworu, składający się z całkowitej liczby taktów (zwykle 1 lub 4) i zbudowany tak, że jego początek i koniec są podobne. Gdy odtwarzasz taki dźwięk raz za razem, słuchacz odnosi wrażenie, że jest to ciągły utwór (a już po chwili, że się jakoś często powtarza). Aby zobaczyć, jak brzmi taki zapętłony fragment, otwórz i posłuchaj pliku *Track 05 Game_AllParts.wav*¹, zapisanego na płycie CD-ROM w podkatalogu *Rozdział_13_14_15/Game/*.



Gotowe pętle dźwiękowe można również znaleźć w Internecie. Jednak ich jakość jest czasami wątpliwa, poza tym często są obwarowane ograniczeniami związanymi z prawami autorskimi. Jedną z moich ulubionych witryn jest www.flashharmonics.com, która udostępnia pętle przygotowane specjalnie dla Flasha. Zestaw pętli dla typowej multimedialnej witryny sieciowej można tam nabyć za około 10 \$. Komercyjne pętle i bridge'e (za moment powiemy, co to takiego) znajdziesz pod adresami:

www.sonicfoundry.com/loop_libraries

www.acidplanet.com

Pętle są powszechnie uważane za bardzo dobre rozwiązanie dla bogatych projektów multimedialnych. Używając kilku pętli, można utworzyć długą kompozycję dźwiękową, której wymagania dotyczące przepustowości łączą się znikome, ponieważ te same dane używane wielokrotnie są przesyłane tylko jeden raz. Dzięki temu możesz użyć bogatszych materiałów wizualnych i lepiej wykorzystać dostępne pasmo. Z tego względu pętle są używane praktycznie wszędzie, gdzie pojawiają się ścieżki dźwiękowe. Jednak mają one pewien mankament — bardzo szybko ich powtarzalność staje się ewidentna, nawet jeśli są dobrze zrobione. Z tego powodu mogą służyć wyłącznie jako muzyka w tle; ich powtarzalna natura sprawia, że trudno ich użyć jako pełnowartościowych utworów muzycznych.

¹ Aby rzeczywiście usłyszeć efekt zapętlenia, otwórz plik w programie, który potrafi odtworzyć ten plik wielokrotnie bez przerw. Niestety, domyślny odtwarzacz w systemie Windows często wstawia krótkie przerwy pomiędzy kolejnymi utworami (lub kolejnymi powtórzeniami tego samego utworu). Jeśli posiadasz program Sound Forge, otwórz plik w tym programie i uruchom jego odtwarzanie za pomocą ikony *Play Looped* — *przyp. tłum.*

Istnieją dwa sposoby rozwiązania tego problemu. Jeden z nich polega na użyciu Flasha jako magnetofonu wielościeżkowego. Możesz przecież utworzyć pętlę zawierającą poszczególne dźwięki lub ścieżki, a następnie miksować je bezpośrednio na listwie czasowej Flasha. Pozwala to wprowadzić znacznie większe zróżnicowanie. Na przykład możesz utworzyć pętlę z sekcją perkusyjną, dodać kilka chwytów gitarowych, kilka sekwencji basu i nieco gitary rytmicznej. Skorzystasz z faktu, że Flash dysponuje ośmioma kanałami dźwięku i może miksować pętle na żywo. Proces ten nie jest często stosowany, ponieważ wymaga dobrego opanowania programów Cubase i Acid (pierwszy z nich już poznaliśmy; drugi poznamy za chwilę).

Drugim sposobem eliminowania ewidentnej powtarzalności pętli jest stosowanie wraz z nimi tak zwanych bridge'ów. Bridge ma taką samą długość jak pętla, lecz nie jest zaprojektowany tak, by go zapętląć. Umieszczamy go pomiędzy dwiema sekwencjami pętli i dzięki temu, ze względu na swą odmienną prowadzą on pewne zróżnicowanie. Bridge'e mają jeszcze jedno, potencjalnie o wiele ciekawsze zastosowanie — jeśli potrafisz utworzyć kod, który będzie wybierał następny bridge i zapętliał różne kombinacje w zależności od tego, co robi użytkownik, jesteś w stanie utworzyć w pełni interaktywną ścieżkę dźwiękową!

Tworzenie bridge'ów wymaga nieco większego uzdolnienia muzycznego i tego samego oprogramowania, którego używamy do tworzenia pętli. Obecnie bridge'e prawdopodobnie w ogóle nie są stosowane na stronach flashowych, zaś interaktywne ścieżki dźwiękowe na bardzo niewielu, ponieważ umiejętności wydajnego wykorzystywania funkcji dźwiękowych Flasha są niewielkie.

Aby doprowadzić do harmonijnej współpracy pętli i bridge'ów, musimy wykonać kilka operacji. Przede wszystkim konieczne jest wycięcie odpowiednich fragmentów z oryginalnej kompozycji. Muszą one być mniej więcej tych samych rozmiarów. Zdecydowałem się na utworzenie fragmentów 4-taktowych, ponieważ 1-taktowe pętle brzmią zbyt mechanicznie. Następnie trzeba przyciąć je do dokładnie takiej samej długości, do czego użyjemy programu Sonic Foundry Acid.

Zanim zaimportujemy pliki dźwiękowe do Flasha, ograniczymy ich zakres częstotliwości za pomocą filtra dolnoprzepustowego. Jest to konieczne, ponieważ we Flashu użyjemy znacznie niższych gęstości strumienia bitowego MP3, niż użyliśmy wcześniej w samodzielnych plikach MP3. Tak znaczna redukcja gęstości strumienia bitowego prowadzi do powstawania metalicznego przydźwięku, wynikającego z niewystarczającej częstotliwości próbkowania. Odfiltrowanie wysokich częstotliwości eliminuje źródło problemu. Proces ten jest potrzebny w przypadku wszystkich importowanych do Flasha plików dźwiękowych, również tych poddanych masteringowi. Większość projektantów flashowych rozwiązuje ten problem inaczej — zwiększając gęstość strumienia bitowego dźwięku MP3; jednak to rozwiązanie jest w oczywisty sposób niedoskonałe, ponieważ zwiększa rozmiar pliku. W procesie tym skorzystamy z programu CoolEdit 2000, ponieważ trzeba go przeprowadzić we wszystkich witrynach korzystających z dźwięku, zaś większość projektantów nie jest tak zaangażowanych w sprawę dźwięku, by kupować programy Sound Forge czy Acid z powodu tej jednej operacji.

Tworzenie pętli i bridge'ów

Utworzenie zestawu pętli i bridge'ów wymaga użycia programu Acid firmy Sonic Foundry oraz pliku MIDI w formacie Cubase, zawierającego oryginalną kompozycję. Istnieje darmowa wersja programu Acid, nosząca nazwę Acid Xpress, do której instrukcja obsługi jest dostępna w postaci pliku PDF. Ta sama instrukcja dotyczy też pozostałych wersji programu Acid (Pro, Music, Style, Xpress); informacje specyficzne dla konkretnych wersji są wyraźnie zaznaczone².

Za pomocą programu Cubase utworzymy nowy zestaw plików WAV, pozbawionych efektów wokalnych. Aby to zrobić, wycisz ścieżki zawierające efekty wokalne (za pomocą funkcji *Mute* — kliknij pole w kolumnie oznaczonej literą *M*). Jeśli korzystasz z własnego utworu, wycisz ścieżki, których nie chcesz umieszczać w interaktywnym tle dźwiękowym (rysunek 14.20). Nagraj potrzebne fragmenty na magnetofon DAT, podobnie jak w poprzednim rozdziale. Następnie poddaj je masteringowi, tak jak zrobiliśmy we wcześniejszej części tego rozdziału.

Rysunek 14.20.



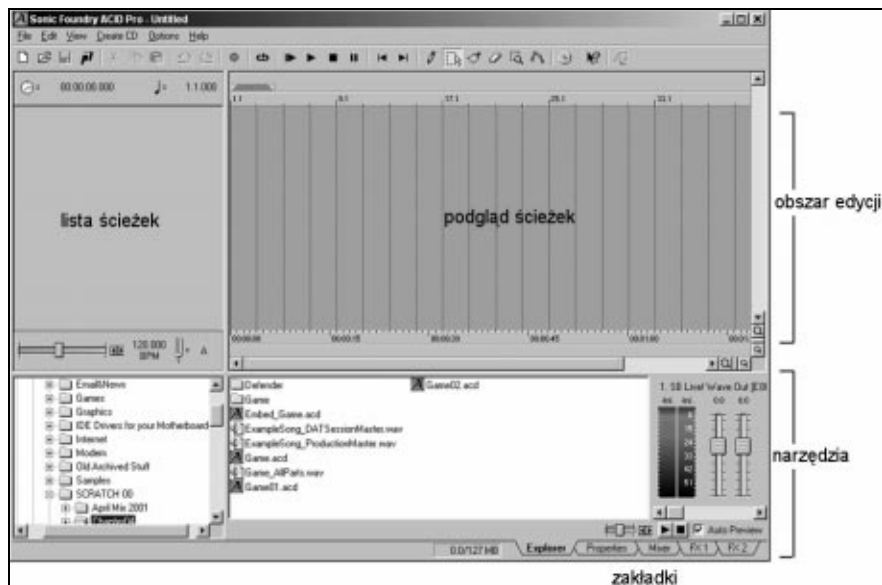
Jeśli zaś chcesz utworzyć indywidualne ścieżki i miksować kompozycję bezpośrednio na listwie czasowej Flasha, nagraj po kolei każdą ścieżkę, wyciszając pozostałe, i powtórz ten proces dla wszystkich ścieżek, których chcesz użyć. W każdym razie poszczególne fragmenty powinny mieć długość czterech pełnych taktów. Zarejestrowane przez nas fragmenty mieszczą się w pliku *Game_AllParts.wav*.

W następnym etapie użyjemy programu Acid Pro, choć do tego samego celu mógłbyś użyć również programu Sound Forge. Program Acid jest o tyle wygodniejszy, że ma bardziej rozwinięte funkcje związane z edycją utworów (na przykład jego listwa czasowa posiada ścieżki i zaawansowane funkcje wyrównywania pętli); z kolei program Sound Forge lepiej nadaje się do tworzenia wszelkiego rodzaju efektów dźwiękowych.

² Oczywiście dokument ten jest w języku angielskim — *przyp. tłum.*

Główny ekran programu Acid jest podzielony na cztery pola. Dwa górne służą edycji ścieżek (jest to odpowiednio lista oraz podgląd ścieżek); dwa dolne zawierają narzędzia ułatwiające proces edycji. Narzędzia różnych kategorii mieszczą się na oddzielnych zakładkach (rysunek 14.21):

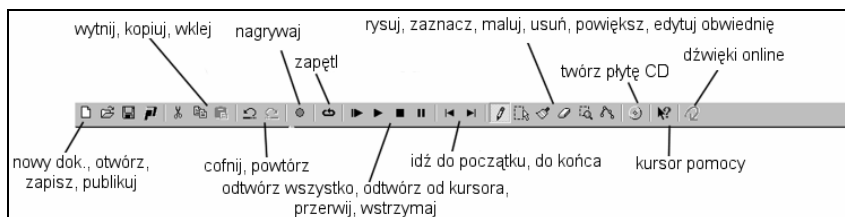
Rysunek 14.21.



- ◆ *Explorer* — ta zakładka ułatwia wybieranie plików (działa podobnie jak Eksplorator w systemie Windows).
- ◆ *Properties* — ta zakładka zawiera właściwości poszczególnych ścieżek. Umożliwia rozciąganie pętli w ścieżkach i ich transpozycję. Funkcje te są szczególnie przydatne w sytuacji, gdy chcesz dopasować ścieżkę lub pętlę wokalu do pozostałych części kompozycji.
- ◆ *Mixer* — ta zakładka umożliwia sterowanie głośnością wszystkich urządzeń dźwiękowych w systemie (nawet modemu, którego raczej nie powinieneś uznawać za instrument godny uwagi).
- ◆ *FX1, FX2* — te zakładki pojawiają się w sytuacji, gdy masz zainstalowane jakieś moduły rozszerzające z efektami.

Ikony na pasku narzędzi w większości przypadków są intuicyjne (rysunek 14.22). Omówimy tylko dwie, które mogą budzić wątpliwości.

Rysunek 14.22.



- ♦ Ikona opisana jako „publikuj” umożliwia wyeksportowanie gotowego utworu i umieszczenie go bezpośrednio na witrynie firmy Sonic Foundry.
- ♦ Ikona opisana jako „dźwięki online” umożliwia pobieranie przykładowych dźwięków utworzonych specjalnie dla programu Acid.

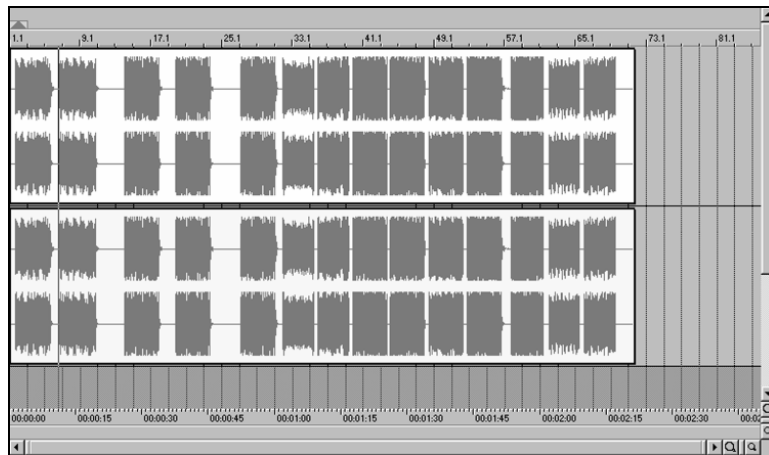
Choć interfejs programu Acid jest stosunkowo prosty, program ten umożliwia łączenie pętli i sampli w taki sam sposób, w jaki program Photoshop umożliwia nakładanie obrazków bitmapowych. Umożliwia wzięcie sampli z jednego nurtu muzyki i przerobienie ich na inny nurt. Z programu tego korzystają zarówno twórcy muzyki pop, jak i artyści tworzący muzykę bardziej eksperymentalną.

Tworzenie pętli za pomocą programu Acid

Jeśli przesłuchałeś plik *Game_AllParts.wav*, wiesz już, że zawiera on kilka 4-taktowych fragmentów utworu. Musimy teraz wyekstrahować je z głównego pliku i edytować tak, aby każdy z nich miał dokładnie taką samą długość jak pozostałe. Dokładność ta powinna być nie mniejsza niż długość pojedynczej klatki w docelowej prezentacji. Prędkość odtwarzania klatek w prezentacji Flasha wyniesie 18 klatek na sekundę, co oznacza, że długość poszczególnych pętli musimy ustalić z dokładnością do jednej osiemnastej sekundy, czyli do 55 milisekund. Inaczej mówiąc, czas trwania pętli może się wahać w granicach $\pm 27,5$ ms.

1. Otwórz plik *Game_AllParts.wav* w programie Acid. Kliknij prawym klawiszem myszy ikonę reprezentującą ścieżkę i z podręcznego menu, które się pojawi, wybierz polecenie *Duplicate Track*. W programie ujrzysz drugą identyczną ścieżkę (rysunek 14.23).

Rysunek 14.23.



2. Rozwiń menu *Options* i upewnij się, że funkcja wyrównania (*Snap* lub *Enable Snap To*, w zależności od wersji programu) jest wyłączona. Zamierzamy umieszczać kursor w określonych miejscach przebiegów dźwiękowych, a ta funkcja by nam to uniemożliwiła.

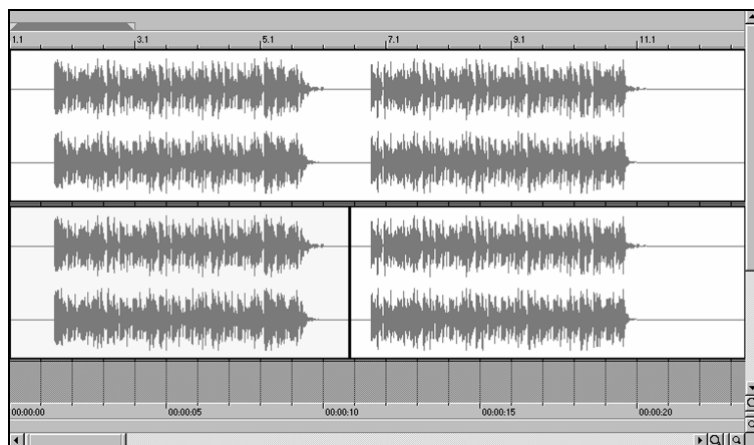
3. Linie kursora możesz przemieszczać, klikając szary poziomy pasek, znajdujący się bezpośrednio nad poziomym paskiem przewijania, i przeciągając kursorem myszy. Umieść linię kursora pomiędzy pierwszą a drugą pętlą. Każda pętla jest reprezentowana na rysunku przez pojedynczy blok dużych wartości przebiegu. Za pomocą ikon lup możesz powiększać obszar podglądu ścieżek, zarówno w poziomie, jak i w pionie. Ikony te znajdują się w prawym dolnym narożniku obszaru podglądu, obok pasków przewijania.
4. Gdy umieścisz kursor pomiędzy pętlami, kliknij zduplikowaną ścieżkę prawym klawiszem myszy i wybierz polecenie *Split At Cursor* (rysunek 14.24). Możesz również nacisnąć klawisz *S* na klawiaturze. Tym sposobem rozdzielisz ścieżkę we wskazanym miejscu na dwie części.

Rysunek 14.24.



5. Rysunek 14.25 przedstawia rozdzieloną ścieżkę. Zaznacz jej lewą część i usuń ją. Zaznacz pozostałą część i przeciągnij ją na początek ścieżki.

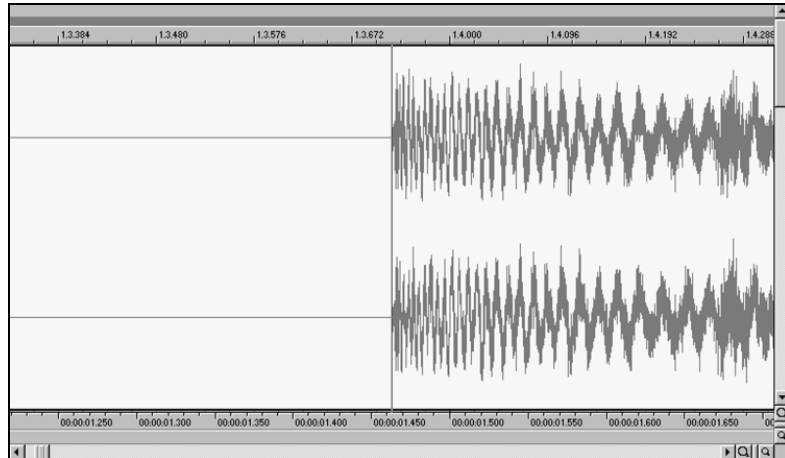
Rysunek 14.25.



6. W tym momencie górna ścieżka rozpoczyna się od pętli numer 1, natomiast dolna pętla rozpoczyna się od pętli numer 2. Powtórz cały proces, tak aby powstała trzecia ścieżka, rozpoczynająca się od pętli numer 3. Słowem, zduplikuj drugą ścieżkę, a następnie odetnij w niej pierwszą widoczną pętlę (czyli pętlę numer 2) i usuń ją; wreszcie przeciągnij pozostałą część ścieżki na początek.
7. Ponownie powtórz cały proces, tworząc w ten sam sposób czwartą ścieżkę, rozpoczynającą się od pętli numer 4. W taki sam sposób rozdziel wszystkie pozostałe pętli w kolejnych ścieżkach, tak aby każda następna ścieżka rozpoczynała się pętlą o kolejnym numerze.

8. Poszczególne pętle są już oddzielone od siebie, jednak przed każdą z nich występuje fragment pustego przebiegu (rysunek 14.26). Wszystkie te puste przebiegi musimy usunąć, tak aby każda pętla rozpoczynała się dokładnie na początku ścieżki. Umieść linię kursora dokładnie na początku pętli i rozdziel ścieżkę tak, jak poprzednio (z podręcznego menu wybierz polecenie *Split at Cursor*). Zaznacz i usuń lewą część ścieżki, a następnie dosuń prawą do początku.

Rysunek 14.26.

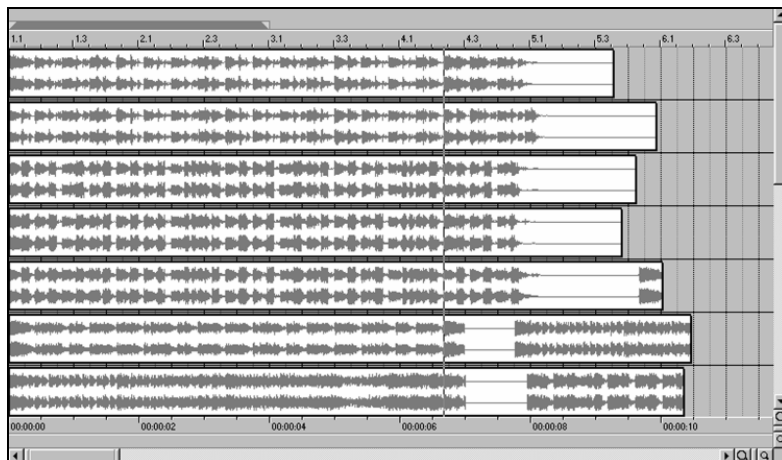


Uwaga

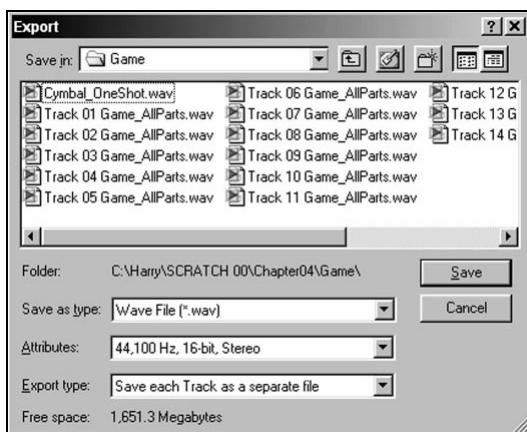
Podkreślam, że w trakcie podziału ścieżki linia kursora musi się znaleźć dokładnie na początku pętli. Im dokładniej ustawisz kursor, tym lepsza będzie uzyskana pętla. Powiększ listwę czasową, aby wskazać początek pętli z dokładnością do pojedynczych milisekund. Dokładność w tym miejscu ułatwi Ci życie na dalszym etapie.

9. Wykonaj ten sam proces w pozostałych ścieżkach. Gdy ukończysz tę pracę, każda ścieżka będzie się rozpoczynała od innej pętli, przy czym na jej początku nie będzie żadnych zerowych wartości. Również końce pętli wypadną w tym samym punkcie listwy czasowej (ponieważ wszystkie pętle mają tę samą długość 4 taktów).
10. Zaznacz wszystkie pętle (użyj skrótu klawiszowego *Ctrl+A*) i przesuń linię kursora na koniec pierwszej pętli (powiększ obraz, aby umieścić kursor jak najdokładniej) (rysunek 14.27). Ponownie dokonaj podziału ścieżek (*Split at Cursor*) i usuń fragmenty ścieżek leżące na prawo od punktu podziału. W tym momencie każda ścieżka powinna zawierać odrębną pętlę, przy czym wszystkie pętle powinny mieć dokładnie taką samą długość.
11. Teraz możesz przystąpić do zapisywania ścieżek jako indywidualnych pętli. Kliknij ikonę zapisu na pasku narzędzi w górnej części ekranu. W oknie dialogowym, które się pojawi, z listy *Export type* wybierz opcję *Save each Track as a separate file* (rysunek 14.28). Zapisz pliki z jakością płyty kompaktowej (opcja *44,100 Hz, 16-bit, Stereo*). Program Acid dołączy na początku podanej nazwy pliku tekst *Track xx*, aby odróżnić poszczególne pliki (zamiast liter *xx* wstawi numer ścieżki).
12. Dla absolutnej pewności sprawdź rozmiary uzyskanych plików. Wszystkie pliki powinny mieć dokładnie taki sam rozmiar, ponieważ każdy z nich mieści pętlę o dokładnie takim samym czasie trwania, a pliki nie zostały poddane kompresji.

Rysunek 14.27.



Rysunek 14.28.



Uzyskane przez nas pliki z pętlami zapisaliśmy w podkatalogu *Rozdział_13_14_15/Game/*; noszą one nazwy *Track 01 Game_AllParts.wav* do *Track 14 Game_AllParts.wav*. Niektóre z tych pętli są do siebie bardzo podobne. Było to naszym zamiarem — warto utworzyć kilka podobnych wersji, ponieważ czasem sąsiadujące ze sobą pętli nie za bardzo do siebie pasują. Zamiast wykonywać całą pracę w programie Acid od początku, można wówczas spróbować dopasować inną, podobną wersję pętli. Problemom tym przyjrzymy się w dalszej części, gdy będziemy umieszczali pętli w prezentacji Flasha. Ponadto niektóre z wygenerowanych plików nie będą zapętlone, lecz będą stanowiły bridge'e pomiędzy dwiema sekcjami pętli.

Jeśli przysłuchasz się tym dźwiękom, zauważysz, że niektóre zawierają na początku głośny dźwięk talerza. W ten sposób można najłatwiej łączyć sąsiadujące ze sobą pętli — nikt wówczas nie zauważy ewentualnych niedociągnięć w łączeniu końca poprzedniej pętli z początkiem następnej! Jeśli wersja pętli bez talerza nie za bardzo pasuje w konkretnym miejscu, zastąp ją wersją z talerzem; jednak nie przesadzaj, ponieważ dźwięk ten jest bardzo charakterystyczny i gdy będzie występował zbyt często, słuchacz szybko znudzi się wyraźnie podkreśloną powtarzalnością.

Przygotowywanie pętli na potrzeby Flasha

Utworzyliśmy już pętle i bridge'e, więc możemy przystąpić do umieszczania ich w prezentacji Flasha. Zanim jednak zaimportujemy pliki do Flasha, omówmy zniekształcenie sygnału wynikające z obniżenia częstotliwości próbkowania. Nawet jeśli tworzysz materiały dla witryny przeznaczonej dla użytkowników łączy szerokopasmowych, musisz znacząco obniżyć ilość danych opisujących dźwięk.

Drastyczne obniżenie częstotliwości próbkowania prowadzi do powstawania charakterystycznego metalicznego przydźwięku — powstaje on wówczas, gdy dźwięk zawiera harmoniczne o częstotliwości wyższej niż połowa częstotliwości próbkowania. Choć samych tych dźwięków nie sposób odtworzyć po obniżeniu częstotliwości próbkowania, prowadzą one do zniekształcenia dźwięków o niższych częstotliwościach. Ponadto może powstawać niepożądany efekt aliasingu (więcej informacji na ten temat w dodatku na końcu książki).

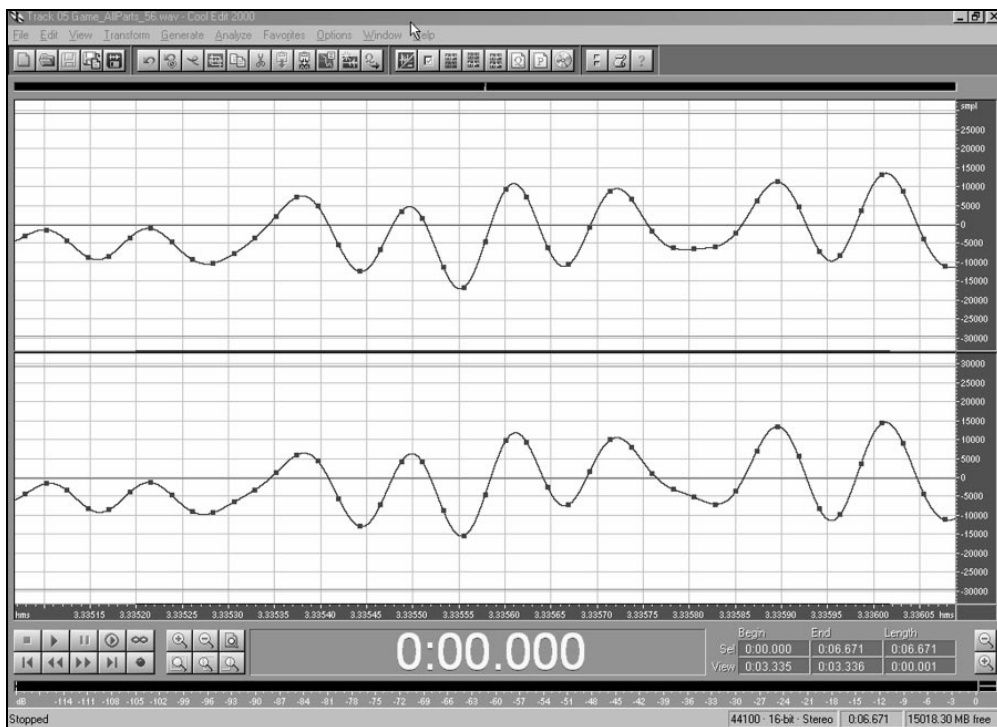
Jest to wystarczający argument, by pozbyć się wysokich częstotliwości przed przystąpieniem do redukcji rozmiaru pliku. Jak to wygląda w praktyce? Przed zaimportowaniem plików WAV do Flasha musisz zdecydować, jak i do jakiej wartości zredukujesz częstotliwość próbkowania podczas kompresji MP3. Znając docelową częstotliwość próbkowania, odfiltruj wszystkie częstotliwości powyżej tak wyznaczonego progu.

Większość projektantów sieciowych zaniedbuje ten ważny krok optymalizacyjny i rozwiązuje problem, decydując się na mniej wydajną kompresję MP3. Dzięki filtracji dolnoprzepustowej możesz zredukować powstające przydźwięki i zaoszczędzić tym sposobem cenne kilobajty. Niektóre programy konwertujące dźwięk do formatu MP3 samodzielnie wykonują filtrację, lecz Flash tego nie robi.

Filtracja dolnoprzepustowa

Aby utworzyć filtr, za pomocą którego przygotujemy pętle na potrzeby Flasha, użyjemy programu CoolEdit 2000. Jego funkcje są podobne do programu Sound Forge, choć istnieją pewne różnice w budowie interfejsu (rysunek 14.29). Przede wszystkim wskaźnik poziomu sygnału znajduje się w dolnej części ekranu, a nie po jego lewej stronie. Poza tym bezpośrednią edycję przebiegu możesz w tym programie przeprowadzić, nie rysując ołówkiem, lecz przeciągając punkty kontrolne, co jest znacznie wygodniejsze i zapewnia większą dokładność edycji. Proces filtracji pokażemy na przykładzie pętli zapisanej w pliku *Track 05 Game_AllParts.wav*.

Zamierzamy utworzyć filtr dolnoprzepustowy. Jest to filtr, który przepuszcza wszystkie częstotliwości mniejsze od wyznaczonej częstotliwości granicznej oraz blokuje wszystkie wyższe częstotliwości. W projekcie Flasha użyjemy gęstości strumienia danych 56 kilobitów na sekundę — jest to optymalna wartość dla użytkowników dysponujących łączem o średniej lub dużej przepustowości, gdy oprócz danych dźwiękowych równolegle są przesyłane strumienie danych dotyczących obrazu. Na podstawie założonej gęstości strumienia danych wyznaczmy teraz częstotliwość graniczną (fachowo zwaną częstotliwością obciążenia) filtra dolnoprzepustowego.



Rysunek 14.29.

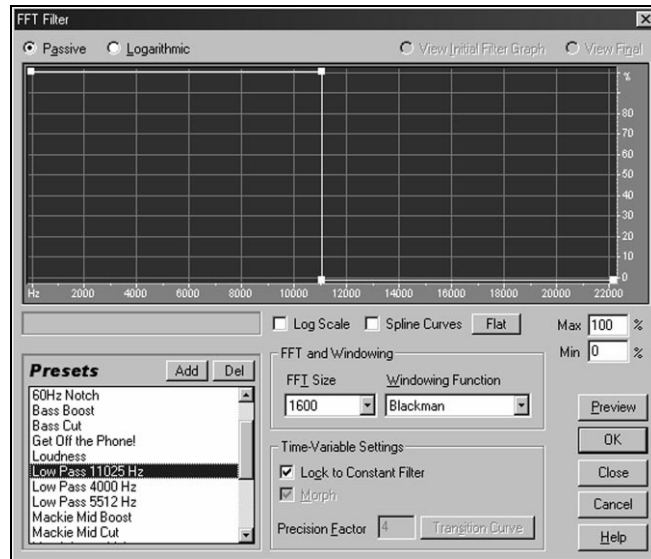
Niestety, gęstość strumienia bitowego nie jest bezpośrednio związana z przenoszoną częstotliwością (choć związek taki istnieje i jest wymuszony przez ograniczoną ilość danych — co potwierdzają eksperymenty). Dlatego dla ułatwienia przygotowaliśmy tabelę z eksperymentalnie wyznaczonymi częstotliwościami obcięcia filtra dolnoprzepustowego, jakie należy przyjąć dla poszczególnych gęstości strumienia danych. Eksperyment polegał na użyciu specjalnie skonstruowanego dźwięku, stanowiącego „eksplozję częstotliwości” z zakresu od 1000 Hz do 22000 Hz. Dźwięk ten poddaliśmy we Flashu kompresji MP3 z różnymi gęstościami strumienia bitowego i sprawdzaliśmy widmo wynikowego dźwięku. Częstotliwości odcięcia, podane w tabeli 14.1, to częstotliwości graniczne; wszystkie częstotliwości od nich wyższe powinny zostać wyeliminowane ze ścieżki dźwiękowej przed poddaniem jej kompresji MP3 z określoną gęstością strumienia bitowego. Tym sposobem unikniesz nieprzyjemnego przydźwięku powstającego przy obniżeniu częstotliwości próbkowania. Ponieważ dźwięki o częstotliwości powyżej 16 kHz są trudne do odróżnienia od powstających przydźwięków (a dla wielu osób po prostu niesłyszalne), w przypadku kompresji MP3 z gęstością strumienia 96 kb/s można zrezygnować z filtracji dolnoprzepustowej.

1. Uruchom program CoolEdit. Jeśli korzystasz z wersji darmowej, możesz korzystać jednocześnie z ograniczonej liczby funkcji. W tym ćwiczeniu będzie potrzebny filtr *FFT Filter*.
2. Otwórz plik dźwiękowy za pomocą polecenia *File/Open*, a następnie wybierz polecenie *Transform/Filters/FFT Filters*. Pojawi się okno dialogowe *FFT Filter* (rysunek 14.30).

Tabela 14.1.

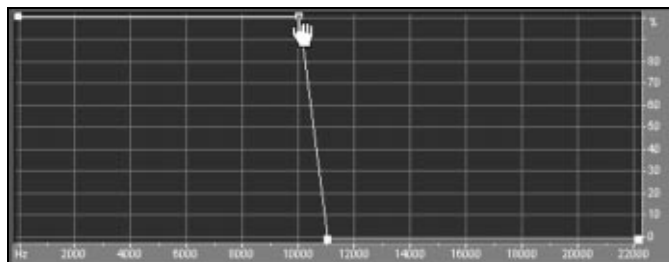
Gęstość strumienia bitowego MP3	Częstotliwość odcięcia filtra dolnoprzepustowego
8 kb/s	3000 Hz
16 kb/s	5500 Hz
20 kb/s	6000 Hz
24 kb/s	6000 Hz
32 kb/s	8000 Hz
48 kb/s	10 000 Hz
56 kb/s	10 000 Hz
64 kb/s	12 000 Hz
80 kb/s	16 000 Hz
96 kb/s	16 000 Hz

Rysunek 14.30.



3. W oknie tym możesz tworzyć filtry, definiując ich charakterystykę częstotliwościową na wykresie. Aby ułatwić sobie życie, na liście *Presets* wybierz gotowe ustawienie, które jest bliskie założonej częstotliwości odcięcia. W naszym przypadku jest to opcja *Low Pass 11025 Hz*. Na wykresie pojawi się charakterystyka częstotliwościowa filtra dolnoprzepustowego o częstotliwości odcięcia 11025 Hz.
4. Musimy ustawić częstotliwość odcięcia filtra na 10000 Hz. Zaznacz górny punkt kontrolny wykresu i przeciągnij go, tak aby znalazł się na linii odpowiadającej częstotliwości 1000 Hz (rysunek 14.31). W niewielkim polu tekstowym nad listą *Presets* możesz odczytać dokładną częstotliwość, jaką ustawiłeś. Jeśli nie potrafisz ustawić dokładnie częstotliwości 10000 Hz, lepiej ustaw za wysoką, niż za niską.
5. Kliknij przycisk *Preview*, aby podслушать rezultat filtracji sygnału dźwiękowego. Zmieniając częstotliwość odcięcia w zakresie od 3000 Hz do 14000 Hz, możesz

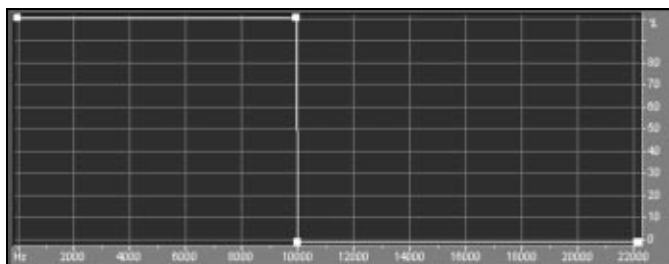
Rysunek 14.31.



sprawdzić, jak zmienia się jakość dźwięku przy gęstościach strumienia zmieniających się we Flashu odpowiednio od 8 kb/s do 112 kb/s. Jest to teoretyczna maksymalna jakość, jaką możesz uzyskać tylko przy dobrze nagrany i zmastrowany dźwięku, który przed kompresją MP3 dodatkowo przefiltrujesz za pomocą odpowiedniego filtra dolnoprzepustowego. Zaimportowanie nieprzefiltrowanego pliku dźwiękowego da w rezultacie dźwięk o znacznie niższej jakości, w związku z szumem kwantyzacji i różnego rodzaju przydźwiękami.

6. W podobny sposób przeciągnij dolny punkt kontrolny na częstotliwości 10000 Hz. Program uniemożliwi przeciągnięcie dolnego punktu w lewo poza górny punkt (rysunek 14.32).

Rysunek 14.32.



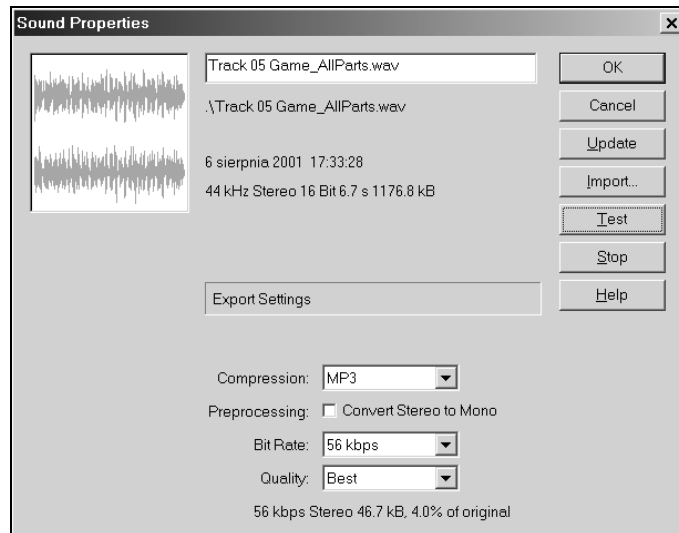
7. Filtr o tak zdefiniowanej charakterystyce bez wątpienia przyda ci się w przyszłości, dlatego zapisz go, klikając przycisk *Add*; w oknie *New Preset Name* wpisz nazwę, która pojawi się na liście *Presets*. Ja nadałem temu filtrowi nazwę *MP3 Anti Alias (48-56kbps)*.
8. Kliknij przycisk *OK*, aby przefiltrować dźwięk. Zapisz rezultat do pliku. Możesz teraz zapisać plik dźwiękowy z niższą częstotliwością próbkowania, ponieważ usunęliśmy z niego wysokie częstotliwości, jednak nie jest to konieczne, ponieważ w dzisiejszych czasach brak miejsca na dysku twardym raczej jest rzadko spotykanym problemem.
9. Zapisz wynikowy plik pod nazwą *Track 5 Game_AllParts_56.wav*. Ten plik można już importować do Flasha.
10. Różnicę pomiędzy przefiltrowaną i nieprzefiltrowaną wersją pliku możesz usłyszeć, przesłuchując rezultaty kompresji obu wersji we Flashu. Uruchom zatem Flasha i zaimportuj obie wersje pliku dźwiękowego (*Track 5 Game_AllParts.wav* oraz *Track 5 Game_AllParts_56.wav*). Obie wersje powinny pojawić się w bibliotece (rysunek 14.33).

Rysunek 14.33.



11. Po kolei dla każdej wersji pliku WAV kliknij dwukrotnie ikonę obok jej nazwy, aby otworzyć okno *Sound Properties*. W polu *Compression* wybierz format MP3 i w polu *Bit Rate* ustaw gęstość strumienia bitowego na 56 kbps (rysunek 14.34). Wyłącz opcję *Convert Stereo to Mono*. Konwersja dźwięku stereofonicznego na monofoniczny nie przyniesie redukcji szumu przy tej gęstości strumienia — schemat kompresji MP3 zawiera na tyle dobry algorytm kompresji panoramy stereo, że konwersja ta ma znaczenie tylko w przypadku niskich gęstości strumienia danych.

Rysunek 14.34.

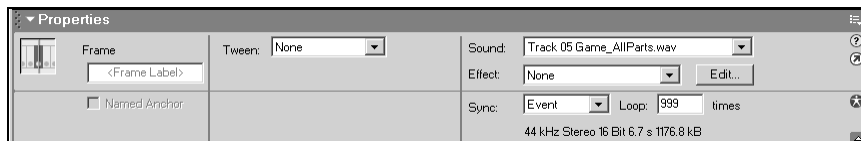


12. Ponieważ do tej pory zakładaliśmy, że użyjemy gęstości strumienia bitowego 56 kb/s, teraz nie powinniśmy ustawiać innej gęstości. Specjalnie dla tej wartości parametru *Bit Rate* przefiltrowaliśmy plik, zatem ustawienie innej gęstości nie wpłynęłoby na poprawę jakości (jeśli mimo tego chcesz użyć innej gęstości strumienia bitowego, powinniśmy ponownie przefiltrować plik, stosując filtr dolnoprzepustowy z odpowiednio dobraną częstotliwością odcięcia).
13. Ustaw parametr *Quality* (jakość) na opcji *Best* (najlepsza). Parametr ten określa, jak dokładnie działa algorytm kompresji MP3. Opcja *Best* korzysta z najdokładniejszego algorytmu, który daje najlepszą możliwą jakość przy danej gęstości strumienia danych (i przy danej jakości pliku wejściowego). Dokładność ta jest okupiona dłuższym czasem kompresowania danych dźwiękowych. W fazie

testów możesz ustawić niższą jakość — *Medium* (średnia) lub *Fast* (szybka, czyli najniższa), aby nie czekać za każdym razem na długotrwałą kompresję dźwięku. Jednak przed publikowaniem filmu zawsze ustaw jakość *Best*. Wbrew informacjom podanym w wielu innych źródłach, ustawienie tego parametru nie ma żadnego wpływu na wielkość wynikowego pliku ani wydajność jego odtwarzania.

14. Film Flasha zawiera jedną pustą warstwę z jednym pustym ujęciem kluczowym. Zaznacz to ujęcie i w panelu *Properties* na liście *Sound* wybierz dźwięk *Track 05 Game_AllParts.wav* (rysunek 14.35). W polu *Loop* wpisz liczbę powtórzeń 999. Uruchom film, by posłuchać rezultatu.

Rysunek 14.35.



15. Gdy już posłuchasz skompresowanego dźwięku, wróć do edycji filmu Flasha. Ponownie zaznacz jedyne ujęcie kluczowe i na liście *Sound* wybierz dźwięk *Track 05 Game_AllParts_56.wav*. Z pewnością zauważysz różnicę pomiędzy obiema wersjami dźwięków — dźwięk przefiltrowany po kompresji zawiera wyższe częstotliwości, z kolei dźwięk nieprzefiltrowany brzmi bardziej płytko i niewyraźnie.

Być może dziwi cię taki wynik — skoro przefiltrowaliśmy plik źródłowy, jak to możliwe, byśmy w pliku wynikowym uzyskali więcej wysokich częstotliwości, niż w wersji niefiltrowanej? Rzeczywiście tak jest i wynika to z teorii próbkowania sygnału. Jeśli częstotliwość próbkowania nie jest wystarczająca, by zarejestrować wszystkie harmoniczne sygnału, wyższe harmoniczne interferują z częstotliwością próbkowania, prowadząc do powstawania szumu, przydźwięku lub innych zniekształceń, których nie było w oryginalnym sygnale, a które niewątpliwie negatywnie wpływają na jakość naszej pracy. (Więcej informacji na temat teorii próbkowania znajdziesz w dodatku na końcu książki). Aby wyeliminować te zniekształcenia, musiałbyś zwiększyć częstotliwość próbkowania lub... wyeliminować przyczynę, czyli odfiltrować zbyt wysokie częstotliwości.

W tym i poprzednim rozdziale omówiliśmy wiele aspektów związanych z dźwiękiem, których większość projektantów sieciowych wydaje się nie brać pod uwagę. Przedstawiliśmy techniki, które pozwalają tworzyć dźwięk o optymalnej dla Internetu jakości (choć niektóre z przedstawionych technik przydadzą się również tym muzykom, którzy pragną rozesłać swoją płytę demo do wytwórni płytowych). Jesteśmy już gotowi, by umieścić utworzone ścieżki dźwiękowe w prezentacji flashowej.

Zajmiemy się tym w następnym rozdziale — zaimportujemy nasze pętle i bridge'y do Flasha, aby utworzyć szybką interaktywną ścieżkę dźwiękową, dynamicznie wzbogacającą materiał wizualny prezentacji. Takie zastosowania nieczęsto można spotkać w dzisiejszym Internecie, lecz może to być najbliższa przyszłość rozbudowanych multimediiów sieciowych — w każdym razie, tego nie powinniśmy przegapić.