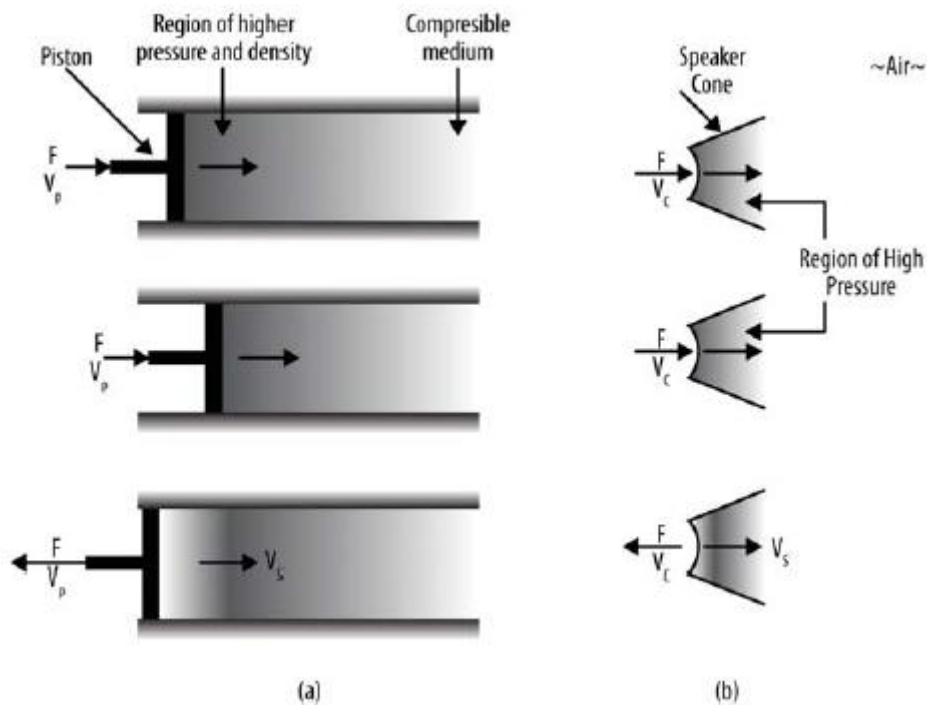


## XXVI. Dźwięk

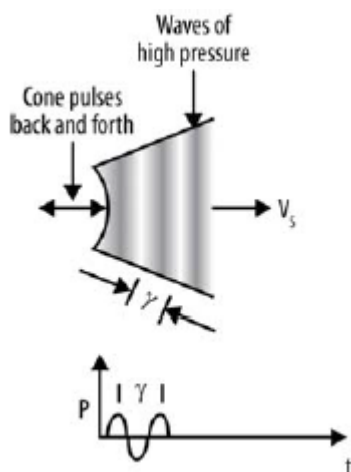
Omówimy podstawową fizykę dźwięku i sposoby przechwytywania efektów dźwiękowych 3D w grach. Odwołujemy się do OpenAL audio API dla niektórych przykładów kodu, ale fizyka, o której rozmawiamy, jest niezależna od konkretnego API. Jeśli jesteś nowy w OpenAL, to w zasadzie OpenGL dla audio. OpenAL wykorzystuje bardzo łatwe do zrozumienia abstrakcje do tworzenia efektów dźwiękowych i obsługuje wszystkie miksy, filtry i syntezę 3D. Zasadniczo tworzysz źródła dźwięku, kojarzysz te źródła z buforami przechowującymi dane dźwiękowe, a następnie manipulujesz tymi źródłami, ustawiając je i ustawiając ich prędkość (wśród innych właściwości). Oczywiście możesz mieć wiele źródeł, ale jest tylko jeden słuchacz. Musisz ustawić właściwości detektora, takie jak pozycja i prędkość słuchacza, aby poprawnie symulować dźwięk 3D. Porozmawiamy o tym więcej w tej części.

### Co to jest dźwięk?

Jeśli przyjrzyj się definicji dźwięku online, otrzymasz odpowiedzi, takie jak dźwięk jest wibracją; odczucie postrzegane przez nasz mózg poprzez stymulację narządów w naszym uchu wewnętrznym; i fluktuację gęstości lub ciśnienia, lub falę, podróżującą przez medium. Więc co to jest? Cóż, to wszystko, a interpretacja, której używasz, zależy od kontekstu, w którym badasz dźwięk. Na przykład, inżynierowie kontroli hałasu, dążący do zminimalizowania hałasu na statkach, koncentrują się na wibracjach propagujących się przez strukturę statku, podczas gdy lekarze martwią się bardziej o biomechanikę naszego ucha wewnętrznego i o tym, jak nasze mózgi interpretują odczucia odbierane przez nasze uszy, a fizycy biorą fundamentalne spojrzenie na wahania gęstości i ciśnienia dzięki materiałom ściśliwym i ich wzajemne oddziaływanie ze sobą i ze środowiskiem. Nie chcemy zasugerować, że każda z tych dyscyplin brzmi tylko w jeden sposób lub w jednym kontekście, ale mówimy, że każda dyscyplina często ma swoją własną perspektywę, priorytety i standardowy język dla tematu. Dla nas, w kontekście gier, dźwięk jest tym, co gracz słyszy przez głośniki lub słuchawki, co pomaga stworzyć wciągające środowisko gry. Jednak, aby tworzyć realistyczne dźwięki, które nadają się do stworzenia wciągającego środowiska, komplementując wciągające wizualizacje i zachowania w grze, musimy zrozumieć fizykę dźwięku, sposób, w jaki go postrzegamy, oraz dźwięk, który mówi nam o jego źródle i środowisko. Biorąc pod uwagę, że jest to książka o fizyce gry, przyjmujemy fundamentalny pogląd na dźwięk, który polega na tym, że dźwięk jest tym, co nasz mózg postrzega, gdy nasze uszy wyczuwają gęstość i wahania ciśnienia w otaczającym nas powietrzu. Te fluktuacje gęstości i ciśnienia są falami i jako takie odniesiemy się do fal dźwiękowych. Przyjrzyjmy się bliżej. Jeśli ściśliwe medium doświadcza zmiany ciśnienia - powiedzmy, ze względu na napędzany tłok - objętość zmieni się, a tym samym jego gęstość ulegnie zmianie. W przypadku napędzanego tłoka, obszar bezpośrednio przed tłokiem będzie najpierw odczuwał kompresję, co skutkuje obszarem zwiększonej gęstości i zwiększonego ciśnienia. Nazywa się to kondensacją. Jeśli chodzi o dźwięk, można myśleć o tym tłoku jako o stożku głośnika. Ten obszar zwiększonej gęstości i ciśnienia będzie się rozprzestrzeniał przez medium, podróżując z prędkością dźwięku w danym medium. Rysunek 26-1 ilustruje tę koncepcję.

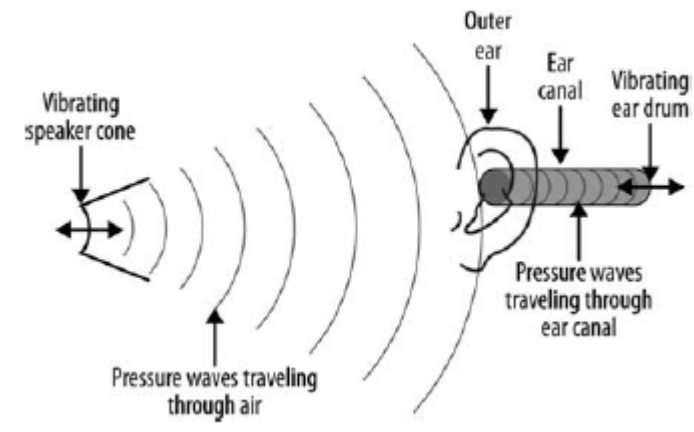


Rysunek 26-1 (a) ilustruje koncepcję tłoka napędzanego, natomiast rysunek 26-1 (b) ilustruje analog głośnika. Gdy tłok lub stożek wypiera płyn (na przykład powietrze), powodując kompresję, a następnie wycofuje się, zostanie utworzony pojedynczy obszar wysokiego ciśnienia, a następnie obszar niskiego ciśnienia. Obszar niskiego ciśnienia wynikający z wycofania tłoka nazywany jest rozrzedzeniem. Ta wynikająca z tego samotna fala ciśnienia zostanie przesunięta w powietrzu w prawo na rysunku 26-1 z prędkością odpowiadającą prędkości dźwięku w powietrzu. Jeśli tłok lub stożek głośnika pulsuje w przód i w tył, jak pokazano na rysunku 26-2, utworzona zostanie seria tych regionów wysokiego / niskiego ciśnienia, co spowoduje ciągłą serię fal - propagację fali dźwiękowej w prawo. Długość fali tej fali dźwiękowej (tj. Odległość zmierzona od wartości szczytowej ciśnienia do wartości szczytowej ciśnienia) jest funkcją częstotliwości pulsacji lub wibracji stożka.



Wynikowa częstotliwość fali dźwiękowej jest powiązana z odwrotnością jego długości fali -  $f = 1 / \lambda$ . Wykres przebiegu amplitudy ciśnienia w funkcji czasu dla tego scenariusza został przedstawiony na rysunku 26-2. Zilustrowaliśmy falę ciśnienia jako sinusoidalną falę sinusoidalną, co nie musi mieć miejsca w rzeczywistości, ponieważ dźwięk pochodzący z głośnika może składać się z wielu różnych

składowych fal. Powiemy o tym więcej później. Jedną rzeczą, na którą chcemy zwrócić uwagę, jest to, że fala dźwiękowa jest falą podłużną, a nie falą poprzeczną. W fali poprzecznej przemieszczenie medium z powodu fali jest prostopadłe do kierunku ruchu fali. W fali podłużnej przesunięcie odbywa się wzdłuż kierunku ruchu fali. Obszary o wyższej gęstości i ciśnieniu fali dźwiękowej są spowodowane ścisnieniem ośrodka wzdłuż kierunku ruchu fali. Zatem fale dźwiękowe są falami podłużnymi. Fale dźwiękowe są zatem zmianami gęstości i ciśnienia przechodzącymi przez medium. Ale jak ich słyszymy? W istocie fala ciśnienia, wytworzona przez pewne wibracje mechaniczne, jak w przypadku stożka głośnika, zostaje przekształcona z powrotem w wibracje w naszym uchu wewnętrznym. I ta wibracja zostaje zinterpretowana przez nasze mózgi jako dźwięk - dźwięk, który słyszymy. Rysunek 26-3 ilustruje tę koncepcję.



Nasze uszy zewnętrzne pomagają uchwycić i skierować fale ciśnienia do naszych kanałów ucha. Te fale ciśnienia przemieszczają się wzdłuż kanału słuchowego, uderzając w bębenek bębenkowy, co powoduje wibrację błony bębenkowej. W tym miejscu zmiany ciśnienia zostają przekształcone z powrotem w wibracje mechaniczne. Poza błoną bębenkową biologia i chemia działają magicznie, przekształcając te drgania w impulsy elektryczne, które nasz mózg interpretuje jako dźwięk. Nasze uszy są wystarczająco czułe, aby wykryć fale ciśnienia w zakresie częstotliwości 20 do 20 000 Hz (Hz). (A Herc to jeden cykl na sekundę.) Interpretujemy częstotliwość jako wysokość dźwięku. Dźwięki o wysokich tonach (think tweeters) odpowiadają wysokim częstotliwościom, a niskie dźwięki (think bass) odpowiadają niskim częstotliwościom. Oprócz tonu oczywistą cechą dźwięku, który odbieramy jest jego głośność. Głośność jest związana z amplitudą fali ciśnienia, między innymi czynnikami, takimi jak czas trwania. Często myślimy o głośności pod względem głośności, mocy lub intensywności. Wszystkie te cechy są ze sobą powiązane i możemy napisać różne formuły odnoszące te cechy do innych cech fali dźwiękowej. Fale dźwiękowe mają energię kinetyczną, która jest związana z masą medium zakłócanego przez falę ciśnienia i prędkością, z jaką ta masa jest zakłócona. Moc to szybkość zmiany energii. Intensywność zależy od tego, ile energii przepływa przez dany obszar. Najważniejsze jest to, że im więcej mocy ma dźwięk, lub im bardziej intensywny, tym głośniejszy wydaje ci się, słuchacz. W pewnym momencie dźwięk może być tak intensywny, że powoduje dyskomfort lub ból. Zazwyczaj intensywność mierzona jest w jednostkach decybeli. Decybel reprezentuje intensywność dźwięku w stosunku do jakiegoś standardowego odniesienia, które zazwyczaj przyjmuje się jako natężenie dźwięku odpowiadające progowi słyszalności. Zero decybeli, czyli 0 dB, odpowiada progowi słyszalności. Intensywność jest tak niska, że jej nie słychać. Kiedy dźwięki osiągają około 120-130 dB, zaczynają powodować ból. Wartości intensywności przedstawione w Tabeli 26-1 są typowe i na pewno istnieje duża różnorodność poziomów w zależności od, na przykład, typu samolotu lub osoby, z którą rozmawiasz, lub miękkości głosu osoby szepczącej do Ciebie. To zdrowy rozsądek, ale jeśli piszesz grę, będziesz chciał odzwierciedlić pewien poziom realizmu w intensywności różnych efektów dźwiękowych w grze. Teraz intensywność jest w rzeczywistości skalą logarytmiczną. Ogólnie przyjmuje

się, że dźwięk zmierzony o 10 dB wyższy od drugiego jest uznawany dwukrotnie za głośniejszy. To jest postrzegana głośność. W ten sposób płaczące dziecko jest ponad dwukrotnie głośniejsze niż normalna rozmowa. Rodzice już to wiedzą.

### Charakterystyka i zachowanie fal dźwiękowych

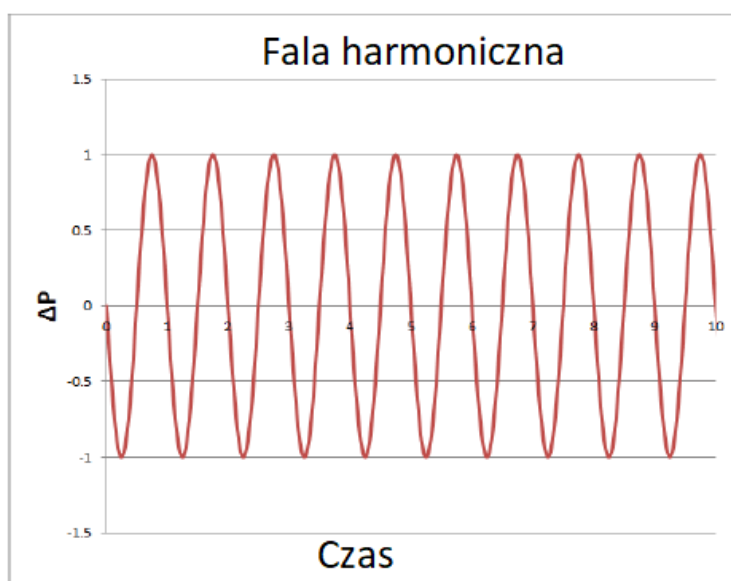
Teraz, gdy ustaliliśmy, czym jest dźwięk, będziemy na ogół odnosić się do dźwięku fal przez pozostałą część tej części. Pamiętaj, że fale dźwiękowe są falami ciśnienia, które nasz mózg interpretuje jako dźwięk. Najważniejsze jest to, że mamy do czynienia z falami, falami podłużnymi. Dlatego możemy użyć zasad mechaniki falowej do opisu fal dźwiękowych. Ponadto, ponieważ fale dźwiękowe (czyli fale ciśnienia) wypierają rzeczywistą masę, mogą wchodzić w interakcje z otoczeniem. Wiemy już, że fale ciśnienia oddziałują z naszymi bębenkami, aby wywołać pewne biochemiczne działanie, powodując, że nasze mózgi interpretują dźwięk. Odwrotnie, środowisko może oddziaływać z falą ciśnienia, aby zmienić jej charakterystykę.

### Fala harmoniczna

Rozważmy jednowymiarową falę ciśnienia harmonicznego, którą można wytworzyć za pomocą napędzanego tłoka pokazanego na rysunku 26-1 (a). Niech kierunek  $x$  odpowiada odległości, dodatni od lewej do prawej. W ten sposób fala z fig. 26-1 (a) przemieszcza się w dodatnim kierunku  $x$ . Niech  $\Delta P$  reprezentuje zmianę ciśnienia z ciśnienia otoczenia w danym momencie. Niech  $A_P$  reprezentuje amplitudę fali ciśnienia. Pamiętaj, że ciśnienie będzie się różnić o pewną wartość większą niż ciśnienie otoczenia, gdy wystąpi kondensacja do pewnej ilości niższej od ciśnienia otoczenia, gdy wystąpi rozrzedzenie. Zakres maksymalnych ciśnień w stosunku do otoczenia to  $-A_P$  do  $+A_P$ . Zakładając falę harmoniczną, możemy napisać:

$$\Delta P = A_P \sin(kx - \omega t - \phi)$$

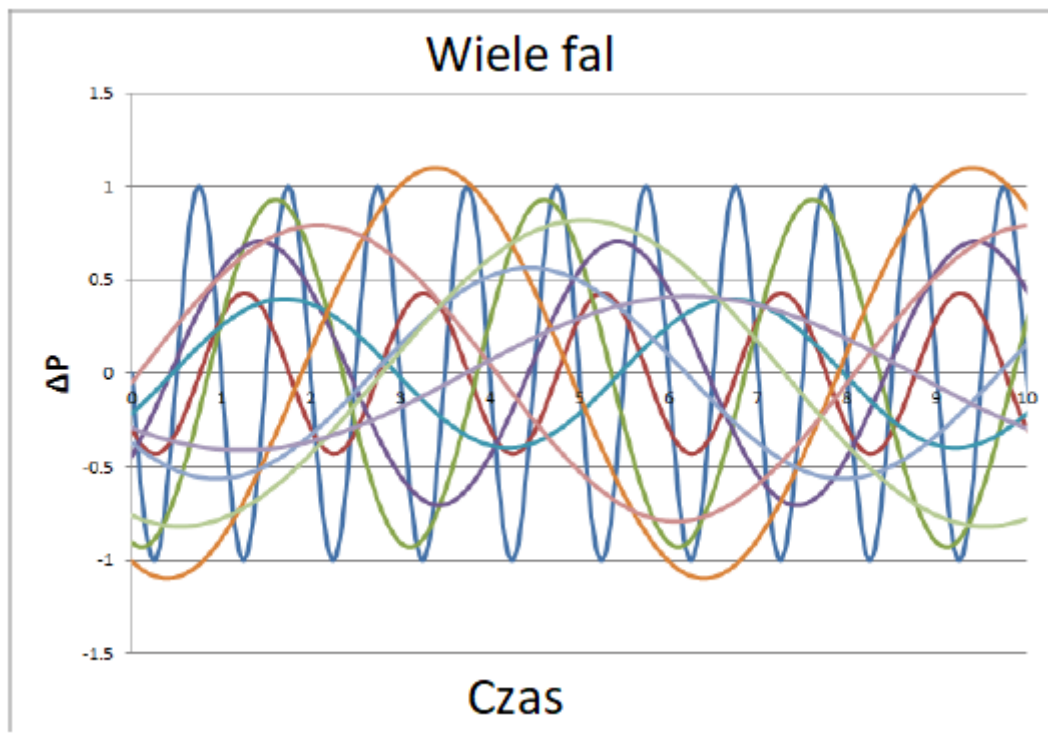
$k$  jest nazywane numerem fali i jest równe  $2\pi/\lambda$ , gdzie  $\lambda$  jest długością fali.  $x$  to współrzędne reprezentujące rozważane stanowisko.  $\omega$  nazywa się kanciastą lub kołową częstotliwością i jest równa  $2\pi f$ , gdzie  $f$  jest częstotliwością fali dźwiękowej.  $t$  reprezentuje czas. Wreszcie,  $\phi$  nazywany jest kątem fazowym, zwanym również przesunięciem fazowym. To reprezentuje przesunięcie fali wzdłuż osi  $X$  w tym przypadku. Dzięki temu równaniu możemy wykreślić, co  $\Delta P$  w konkretnym położeniu  $x$  w funkcji czasu. Rysunek 26-4 ilustruje taki wykres, zakładając, że  $A_P$ ,  $\lambda$  i  $f$  są równe 1, a  $\phi$  jest równe 0.



Fizycznie, gdybyś zmierzył zmianę ciśnienia w czasie,  $x$ , wykres wyglądałby tak, jak pokazano na rysunku 26-4.

### Superpozycja

Ogólnie, fale dźwiękowe nie wyglądają jak czysta fala harmoniczna pokazana na rysunku 26-4 chyba że dźwięk jest czystym tonem. Nie-czysty ton będzie miał inne ruchy na swoim wykresie, wynikające z komponentów na innych częstotliwościach i fazach. Podobnie, jeśli na przykład nagrażasz ciśnienie akustyczne w jednym punkcie pomieszczenia, w którym istnieje wiele źródeł dźwięku, zapis ciśnienia w danym punkcie nie będzie odpowiadał dźwiękowi konkretnego źródła dźwięku. Zamiast tego zapisana historia czasu ciśnienia będzie kombinacją wszystkich obecnych źródeł, a to, co usłyszysz, to połączenie wszystkich źródeł dźwięku. Dobrym przybliżeniem tego, w jaki sposób te różne komponenty dźwiękowe się łączą, jest po prostu zsumowanie wyników każdego komponentu w danym punkcie. Jest to zasada superpozycji. Rysunek 26-5 pokazuje 10 różnych przebiegów, każdy z różnymi amplitudami, częstotliwościami i fazami.



Zasada superpozycji mówi, że możemy dodać wszystkie te przebiegi, aby określić łączny wynik. Rysunek 26-6 pokazuje wynikową falę.



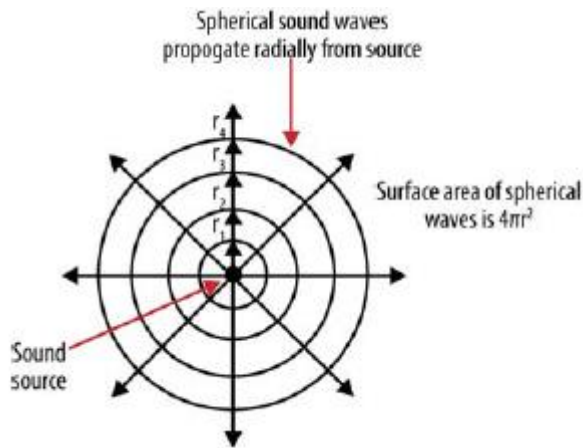
Zwróć uwagę, że poszczególne fale są dodawane algebraicznie. W każdej chwili w czasie niektóre fale wytwarzają dodatnie zmiany ciśnienia, podczas gdy inne powodują zmiany podciśnienia. Oznacza to, że niektóre fale łączą się, aby dokonać większych zmian ciśnienia, ale oznacza to również, że niektóre z nich mogą się sumować w celu dokonania mniejszych zmian ciśnienia. Innymi słowami, fale mogą być konstruktywne lub destrukcyjne. Niektóre fale mogą całkowicie wyeliminować się nawzajem, co jest podstawą technologii redukcji szumów.

### Prędkość dźwięku

Fale dźwiękowe przemieszczają się przez medium z pewną skończoną prędkością, co jest funkcją właściwości elastycznych i bezwładności tego ośrodka. Ogólnie dźwięk podróżuje szybciej w sztywniejszych, mniej ściśliwych mediach niż w bardziej miękkich lub bardziej ściśliwych mediach. Na przykład prędkość dźwięku w powietrzu wynosi około 340 m / s, w zależności od temperatury, wilgotności i innych czynników, ale w wodzie morskiej wynosi około 1500 m / s. Woda jest znacznie mniej ściśliwa niż powietrze. W związku z tym krokiem prędkość dźwięku w bryle takiej jak żelazo wynosi około 5100 m / s. Możesz powiedzieć: "Więc co; dlaczego muszę się martwić szybkością dźwięku w mojej grze?" Cóż, prędkość, z jaką wędrują fale dźwiękowe, mówi nam coś o źródle dźwięku, i możesz wykorzystać te wskazówki w swojej grze, aby wzmocnić wciągającą atmosferę. Załóżmy, że wroga jednostka wystrzeliwuje broń w strzelance 3D, a wróg znajduje się w pewnej odległości od twojego gracza. Gracz powinien zobaczyć błysk wylotowy, zanim usłyszy dźwięk strzelania z pistoletu. Opóźnienie to wynika z faktu, że światło przemieszcza się znacznie szybciej niż dźwięk. Opóźnienie pomiędzy ujrzaniem błysku wylotowego a usłyszeniem strzału daje graczowi trochę dystansu, z którego wróg strzela. W odniesieniu do efektów dźwiękowych 3D, nasze uszy słyszą dźwięki wydobywające się z ukośnego kierunku w nieco innym czasie ze względu na odległość pomiędzy naszymi uszami. Opóźnienie, choć bardzo krótkie, daje nam wskazówki co do kierunku, z którego nadchodzi dźwięk. Powiemy więcej na ten temat w dalszej części. Dodatkowo efekt Dopplera, który omówimy później, jest również funkcją prędkości dźwięku. W OpenAL ustawiasz żądaną prędkość dźwięku za pomocą funkcji `ISpeedOfSound`, przekazując pojedynczy argument zmiennoprzecinkowy reprezentujący prędkość dźwięku. Określona wartość jest zapisywana we właściwości `AL_SPEED_OF_SOUND`. Domyślna wartość to 343,3, czyli prędkość dźwięku w powietrzu przy 20 ° C wyrażona w m / s.

## Oslabienie

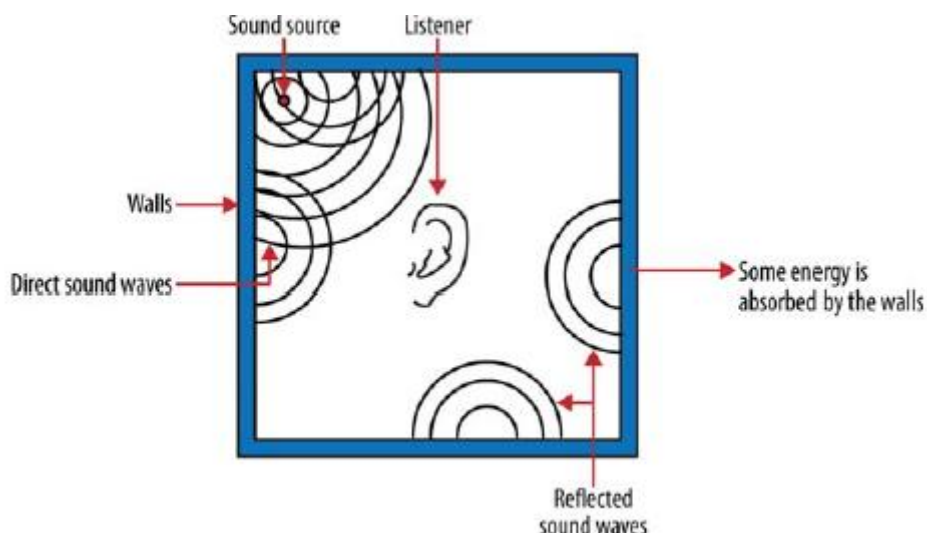
Tłumienie to spadek intensywności dźwięku na odległość. Wcześniej wyjaśnialiśmy, że natężenie dźwięku zależy od tego, ile energii przepływa przez dany obszar. Wyobraź sobie punktowe źródło dźwięku, które wytwarza sferyczne fale ciśnienia, które propagują promieniowo ze źródła. Rysunek 26-7 ilustruje tę koncepcję.



Zakładając, że dźwięk jest generowany ze stałą mocą, widać, że obszar, przez który przepływa ta moc, rośnie wraz ze wzrostem odległości,  $r$ , od źródła. Intensywność jest równa mocy podzielonej przez obszar, a więc intensywność przy promieniu  $r_4$  jest mniejsza niż na powiedzmy  $r_1$ , ponieważ pole powierzchni w  $r_4$  jest większe. Powierzchnia kuli to  $4\pi r^2$ . Nie wchodząc w szczegóły, możemy stwierdzić, że amplituda sferycznej fali dźwiękowej jest odwrotnie proporcjonalna do  $r^2$ . To jest jak dotąd idealne leczenie. W rzeczywistości tłumienie jest również funkcją innych czynników, w tym rozpraszania i pochłaniania fali dźwiękowej, gdy oddziałuje ona z medium i otoczeniem. Możesz modelować tłumienie na wiele sposobów, biorąc pod uwagę różne poziomy szczegółowości przy zwiększaniu kosztów obliczeniowych. Jednak w przypadku gier wystarczające są stosunkowo proste modele oparte na odległości. Tłumienie stanowi kolejną wskazówkę, która mówi nam coś o źródle dźwięku. W swojej grze nie chcesz, aby natężenie lub głośność dźwięku generowanego z dala od odtwarzacza było takie samo jak w przypadku źródła bardzo zbliżonego do odtwarzacza. Tłumienie mówi graczowi o odległości między nim a źródłem dźwięku. OpenAL zawiera kilka różnych modeli opartych na odległości, z których możesz wybierać. Dokumentacja OpenAL opisuje szczegóły każdego z nich, ale model domyślny jest odwrotnym modelem opartym na odległości, w którym wzmocnienie źródła dźwięku jest regulowane odwrotnie proporcjonalnie do odległości od źródła dźwięku. Wzmocnienie jest współczynnikiem wzmocnienia stosowanym do zarejestrowanej amplitudy efektu dźwiękowego, którego używasz. Możesz zmieniać modele odległości w OpenAL za pomocą funkcji `alDistanceModel`.

## Odbicie

Kiedy fale dźwiękowe przechodzące przez jedno medium docierają do innego medium lub obiektu, takiego jak ściana, część pierwotnej fali dźwiękowej odbija się od obiektu, a jej część jest absorbowana przez obiekt (i transmitowana przez niego). W zależności od dyspozycji źródła dźwięku i słuchacza, niektóre fale dźwiękowe docierają do słuchacza przez pewną bezpośrednią ścieżkę. Odbite fale mogą również dotrzeć do słuchacza, chociaż ich energia może zostać zmniejszona po ich interakcji z tym, co odbiły. Rysunek 26-8 ilustruje tę koncepcję, w której niektóre fale dźwiękowe docierają bezpośrednio do słuchacza, a inne docierają do słuchacza po odbiciu od ścian.



Stopień, w jakim odbijają się fale dźwiękowe, ma związek z charakterystyką materiału, z którego odbijają się. Gładkie, twarde powierzchnie będą miały tendencję do odbijania większej ilości energii dźwiękowej, podczas gdy bardziej miękkie, nieregularne powierzchnie będą pochłaniać więcej energii i rozpraszać odbite fale. Te cechy nadają pewną jakość ostatecznemu dźwiękowi, który słyszy słuchacz. Ten sam dźwięk grany w wyłożonej kafelkami łazience będzie miał inną jakość niż w pokoju z dywanem, zasłonami i gobelinami. W łazience dźwięk może brzmieć echem, podczas gdy w pokoju z dywanem może wydawać się wyciszony. Gdzieś pomiędzy tymi dwoma typami pokoi dźwięk może rozbrzmiewać. Pogłos jest postrzeganym przedłużeniem pierwotnego dźwięku z powodu odbić dźwięku w przestrzeni. W twoich grach byłoby niezmiernie kosztowne (mówiąc komputerowo), aby spróbować modelować różne źródła dźwięku, współdziałając ze wszystkimi ścianami i obiektami w dowolnej przestrzeni w grze w czasie rzeczywistym. Takie obliczenia są możliwe i są często stosowane w inżynierii akustycznej i aplikacjach kontroli hałasu, ale znowu jest to zbyt kosztowne dla gry. To, co możesz zrobić, to naśladować refleksyjne lub pogłosowe cechy danej przestrzeni w twoim środowisku gry poprzez dostosowanie pogłosu twoich źródeł dźwięku. Jednym ze sposobów jest rejestrowanie efektów dźwiękowych o jakości, której szukasz, aby reprezentować przestrzeń, w której efekt dźwiękowy byłby zastosowany. Na przykład możesz nagrać echo dźwięków kapiącej wody w kamiennym pomieszczeniu, aby poprawić atmosferę lochu. Alternatywnie, jeśli używasz systemu takiego jak OpenAL i jeśli efekt pogłosu jest dostępny na karcie dźwiękowej, możesz przypisać określone charakterystyki pogłosu poszczególnym źródłom dźwięku, aby naśladować określone środowiska. Tego rodzaju podejście należy do sfery modelowania środowiskowego, a przewodnik OpenAL Effects Extension Guide (część dokumentacji OpenAL) podaje kilka całkiem niezłych wskazówek dotyczących wykorzystania rozszerzeń efektów specjalnych do modelowania środowiskowego.

## Efekt Dopplera

Efekt Dopplera powstaje, gdy występuje względny ruch pomiędzy źródłem dźwięku a słuchaczem. Przejawia się to jako wzrost częstotliwości, gdy źródło i słuchacz zbliżają się do siebie, oraz zmniejszenie częstotliwości, gdy źródło i słuchacz oddalają się od siebie. Na przykład róg zbliżającego się pociągu wydaje się zwiększać wysokość, gdy zbliża się do niego, ale wydaje się zmniejszać wysokość, gdy pociąg mija i odjeżdża. Efekt Dopplera jest bardzo oczywistą wskazówką odnośnie względnego ruchu źródła dźwięku, które można uchwycić w swoich grach. Na przykład możesz modelować dźwięk pędzącego samochodu z efektem Dopplera, uzupełniając wizualne wskazówki samochodu zbliżającego się i przechodzącego przez gracza. To, co się dzieje fizycznie, to częstotliwość spotkań fal dźwiękowych do słuchacza jest zwiększona ze względu na prędkość względną. Zbliżająca się prędkość oznacza, że



słuchacz napotyka więcej fal na jednostkę czasu, co jest słyszane jako częstotliwość wyższa niż częstotliwość źródłowa. Odwrotnie, prędkość odchodząca oznacza, że napotykanym jest mniej fal na jednostkę czasu, która jest słyszana jako niższa częstotliwość. Zakładając nieruchome powietrze, zwiększona częstotliwość słyszalna, gdy źródło dźwięku i słuchacz zbliżają się do siebie, jest określana przez relację:

$$f_h = f [(c + v_l) / (c + v_s)]$$

gdzie  $f_h$  jest częstotliwością słyszaną przez słuchacza,  $c$  jest prędkością dźwięku,  $v_l$  jest prędkością słuchacza, a  $v_s$  jest prędkością źródła. To równanie pokazuje, że częstotliwość słyszana przez słuchacza jest zwiększana proporcjonalnie do stosunku sumy prędkości dźwięku plus względnej prędkości źródła i słuchacza do prędkości dźwięku. Jeśli źródło i słuchacz oddalają się od siebie, to  $v_r$  jest ujemne, a słyszana częstotliwość jest zmniejszana. W swojej grze możesz używać wstępnie nagranych dźwięków z efektami Dopplera do przejeżdżających samochodów lub innych ruchomych źródeł dźwięku; nie będzie jednak możliwe dostosowanie efektu Dopplera do rzeczywistej prędkości względnej między źródłem dźwięku a słuchaczem, jeśli symulujesz obiekt, do którego dźwięk jest dołączony, przy użyciu technik opisanych w tej książce. Twój nagrany dźwięk jest poprawiony. Może to dobrze działać, przy okazji. Mimo to, jeśli korzystasz z systemu audio takiego jak OpenAL, możesz użyć wbudowanych funkcji efektu Dopplera. Zasadniczo, jeśli symulujesz obiekt generujący dźwięk, aktualizujesz jego źródło dźwięku, aby odzwierciedlało prędkość obiektu, a jednocześnie aktualizujesz obiekt słuchacza, aby odzwierciedlić prędkość gracza. OpenAL zajmie się resztą. Pokażemy prosty przykład tego w następnej sekcji.

### **Dźwięk 3D**

W pewnym momencie nie tak dawno temu "dźwięk 3D" został zahipnotyzowany jako kolejna wielka rzecz. Nie ma wątpliwości, że przez długi czas dźwięk gry pozostawał daleko w tyle za możliwościami graficznymi. Prawdą jest również, że dobry dźwięk 3D może pochwalić się dobrą grafiką, pomagając stworzyć bardziej wciągające środowisko gry. Niestety, dużo wczesnego dźwięku 3D po prostu nie było dobry. Sprawy stają się coraz lepsze, a przy użyciu słuchawek i dobra karta dźwiękowa, można wygenerować niesamowite dźwięki 3D. Przy prawidłowym stosowaniu dźwięk 3D może dać graczowi wrażenie, że dźwięki pochodzą z różnych różnych kierunków. Na przykład strzałowi wystrzelonemu zza odtwarzacza towarzyszyłby gracz usłyszawszy dźwięk wystrzału z pistoletu, jakby rzeczywiście dochodził z tyłu. Takie kierunkowe brzmienie naprawdę dodaje do wciągającego wrażenia z gry.

### **Jak słyszymy w 3D**

Dźwięk 3D - a dokładniej nasza umiejętność lokalizowania dźwięku - jest wynikiem złożonej interakcji między źródłem dźwięku a naszym ciałem, nie wspominając o pomieszczeniu lub otoczeniu, w którym się znajdujemy. Ignorowanie interakcji środowiskowych, Rysunek 26-9 ilustruje, w jaki sposób fala dźwiękowa współdziała z własnym ciałem. Jedną z pierwszych rzeczy, które możesz zauważyć, jest to, że nasze uszy są oddzielone pewnym skończonym dystansem. Oznacza to, że dźwięk dochodzący od źródła po prawej stronie dotrze do prawego ucha, zanim dotrze do lewego ucha. Opóźnienie między dźwiękiem docierającym do każdego ucha nazywa się opóźnieniem międzywojowym. Możemy w przybliżeniu obliczyć opóźnienie od dźwięku dochodzącego z boku, biorąc dystans dzielący uszy i dzieląc go przez prędkość dźwięku. W powietrzu i typowym rozmiarze głowy opóźnienie to wynosi około pół milisekundy. Opóźnienie będzie krótsze w zależności od orientacji głowy względem źródła dźwięku. Niezależnie od opóźnienia, nasze mózgi wykorzystują te informacje, aby określić lokalizację, z której dochodzi dźwięk. Dodatkowo, gdy dźwięk dochodzący z prawej strony na rysunku 26-9 dociera do głowy, część energii odbija się od głowy. Odbicia występują również na ramionach i tułowiu. Ponadto, gdy fale dźwiękowe przechodzą przez głowę, mają skłonność do pochylania się wokół niej.

Fale o wyższej częstotliwości są zwykle blokowane przez głowę, a fale o niższej częstotliwości mają tendencję do przechodzenia z niewielkimi przerwami. Wynikowy dźwięk w obszarze cienia za głową jest nieco inny niż źródło ze względu na skuteczne filtrowanie, które nastąpiło w wyniku interakcji z głową. Zauważ również, że orientacja uszu w odniesieniu do źródła dźwięku jest różna, a fale dźwiękowe będą inaczej oddziaływać z ucha i kanału słuchowego z powodu tej różnej orientacji. Jeśli dźwięk dochodzi z góry lub z dołu osoby, oprócz przesunięcia w bok, dźwięk odbija się i dyfunduje wokół różnych części ciała na różne sposoby. Biorąc pod uwagę wszystkie te interakcje, wydaje się, że dźwięk, który słyszymy, jest zupełnie inny niż dźwięk czystego źródła. Różnice mogą nie być tak dramatyczne, ale są wystarczające, aby nasz mózg mógł wykryć wszystkie te sygnały, pozwalając nam zlokalizować źródło dźwięku. Biorąc pod uwagę, że wszyscy mamy różne kształty i rozmiary, nasze mózgi są dostrojone do naszych konkretnych ciał podczas przetwarzania tych wskazówek lokalizacyjnych. Wydaje się, że w przypadku wiarygodnego dźwięku 3D praktycznie niemożliwe jest osiągnięcie w grach ze względu na złożoność dźwięków wchodzących w interakcje ze słuchaczem. Z pewnością nie możesz modelować każdego potencjalnego gracza gry wraz z dźwiękiem gry, aby obliczyć, w jaki sposób wchodzi z sobą w interakcje. To powiedziawszy, jednym z podejść do przechwytywania ważnych wskazówek lokalizacyjnych jest użycie tak zwanych funkcji przeniesienia głowy (HRTF). Jeśli umieścisz mały mikrofon w każdym uchu, a następnie nagrasz dźwięk dochodzący do każdego ucha z jakiegoś znanego źródła, otrzymasz to, co nazywa się nagraniem obustronnym. Innymi słowy, dwa nagrania - po jednym dla każdego ucha - przechwytyują dźwięk otrzymany przez każdego, który, biorąc pod uwagę wszystkie czynniki, które opisaliśmy wcześniej, różni się od siebie. Te dwa nagrania zawierają informacje, których nasz mózg używa, aby pomóc nam zlokalizować dźwięk źródłowy. Teraz, jeśli porównasz te nagrania dwuoszne, przyjmując stosunek każdego z nich do źródła dźwięku, skończysz z czymś, co nazywa się funkcją transferu dla każdego ucha. (Matematyka jest bardziej skomplikowana, niż sugerujemy tutaj.) Są to HRTF. Możesz też uzyskać HRTF dla dźwięku zlokalizowanego w dowolnej pozycji względem słuchacza. Tak więc, nagrania dwuoszne dla źródła zlokalizowanego w określonym miejscu dają parę HRTF. To nie jest tak źle, ale to tylko dla jednego źródła lokalizacji. Jeśli chcesz emulować dźwięk 3D z dowolnego miejsca, potrzebujesz HRTF dla każdej lokalizacji. Oczywiście generowanie HRTF dla każdej możliwej lokalizacji względnej jest niepraktyczne, więc HRTF są zwykle uzyskiwane z dwuosznych zapisów wykonanych w wielu dyskretnych lokalizacjach, aby utworzyć bibliotekę, by tak rzec, funkcji transferu. HRTF są następnie wykorzystywane do wyprowadzania filtrów dla danego dźwięku, który chcesz odtwarzać za pomocą emulacji 3D. Wymagane są dwa filtry - po jednym dla każdego ucha. A HRTF używane do wyprowadzania tych filtrów to te, które są najbliższe lokalizacji źródła dźwięku 3D, które próbujesz naśladować. Dużo pracy wymaga nagrania wszystkich tych nagrań i wyprowadzenia odpowiednich HRTF. Czasami nagrania są wykonywane za pomocą manekina, a czasami używa się prawdziwych ludzi. W obu przypadkach jest mało prawdopodobne, że ty lub twój gracz będziecie przypominać manekina lub człowieka, którego używa się do nagrań i HRTF. Oznacza to, że zszyntetyzowany dźwięk 3D może jedynie przybliżyć wskazówki dla konkretnej osoby.

### **Prosty przykład**

OpenAL pozwala symulować dźwięk 3D za pomocą łatwych w użyciu obiektów źródłowych i słuchających z powiązаныmi właściwościami, takimi jak między innymi pozycja, prędkość i orientacja. Wystarczy skojarzyć dane dźwiękowe ze źródłem i ustawić jego właściwości, pozycję słuchacza, prędkość i orientację. OpenAL zajmie się resztą. Skuteczność wyników zależy od implementacji OpenAL i używanego sprzętu dźwiękowego. OpenAL pozostawia implementację rzeczy takich jak HRTF do sprzętu. Dla celów demonstracyjnych wzięliśmy przykład PlayStatic dostarczony w SDK Creative Labs OpenAL i zmodyfikowaliśmy go nieco, aby źródło dźwięku poruszało się wokół słuchacza. Dodaliśmy

również efekt Dopplera, aby sprawić wrażenie, że źródło porusza się w kierunku lub od słuchacza. Odpowiedni kod jest następujący:

```
int main()
{
    ALuint uiBuffer;
    ALuint uiSource;
    ALint iState;
    // Initialize Framework
    ALFWInit();
    if (!ALFWInitOpenAL())
    {
        ALFWprintf("Failed to initialize OpenAL\n");
        ALFWShutdown();
        return 0;
    }
    // Generate an AL Buffer
    alGenBuffers( 1, &uiBuffer );
    // Load Wave file into OpenAL Buffer
    if (!ALFWLoadWaveToBuffer((char*)ALFWaddMediaPath(TEST_WAVE_FILE), uiBuffer))
    {
        ALFWprintf("Failed to load %s\n", ALFWaddMediaPath(TEST_WAVE_FILE));
    }
    // Specify the location of the Listener
    alListener3f(AL_POSITION, 0, 0, 0);
    // Generate a Source to playback the Buffer
    alGenSources( 1, &uiSource );
    // Attach Source to Buffer
    alSourcei( uiSource, AL_BUFFER, uiBuffer );
    // Set the Doppler effect factor
    alDopplerFactor(10);
    // Initialize variables used to reposition the source
```

```
float x = 75;
float y = 0;
float z = -10;
float dx = -1;
float dy = 0.1;
float dz = 0.25;
// Set Initial Source properties
alSourceci(uiSource, AL_LOOPING, AL_TRUE);
alSource3f(uiSource, AL_POSITION, x, y, z);
alSource3f(uiSource, AL_VELOCITY, dx, dy, dz);
// Play Source
alSourcePlay( uiSource );
do
{
Sleep(100);
if(fabs(x) > 75) dx = -dx;
if(fabs(y) > 5) dy = -dy;
if(fabs(z) > 10) dz = -dz;
alSource3f(uiSource, AL_VELOCITY, dx, dy, dz);
x += dx;
y += dy;
z += dz;
alSource3f(uiSource, AL_POSITION, x, y, z);
// Get Source State
alGetSourceci( uiSource, AL_SOURCE_STATE, &iState);
} while (iState == AL_PLAYING);
// Clean up by deleting Source(s) and Buffer(s)
alSourceStop(uiSource);
alDeleteSources(1, &uiSource);
alDeleteBuffers(1, &uiBuffer);
ALFWShutdownOpenAL();
```

```
ALFWShutdown();  
  
return 0;  
  
}
```

W tej demonstracji nie ma nic nadzwyczajnego, a w rzeczywistości nie ma grafiki. Program działa w oknie konsoli. Jest to jednak w porządku, ponieważ naprawdę potrzebujesz uszu, a nie oczu, aby docenić tę demonstrację. Pamiętaj, aby używać słuchawek, jeśli sam je przetestujesz. Efekt 3D jest znacznie lepszy w przypadku słuchawek. Wiersze kodu w obrębie głównej () aż do komentarza Określ lokalizację odbiornika to po prostu wywołania inicjalizacyjne OpenAL wymagane do skonfigurowania architektury i skojarzenia pliku dźwiękowego z buforem dźwiękowym, który będzie przechowywać dane dźwiękowe do późniejszego odtwarzania. Następny wiersz kodu po powyższym komentarzu określa położenie detektora. Określamy lokalizację odbiornika u źródła. W grze ustawisz lokalizację słuchacza na lokalizację gracza, gdy gracz porusza się po twoim świecie gry. W tym przykładzie słuchacz zostaje ustawiony. Następnie tworzone jest źródło i skojarzone z wcześniej utworzonym buforem dźwiękowym. Ponieważ chcemy uwzględnić efekt Dopplera, ustawiamy współczynnik Dopplera na 10. Wartość domyślna to 1, ale wzmocniliśmy ją, aby wzmocnić efekt. Następnie tworzymy sześć nowych zmiennych lokalnych do przechowywania współrzędnych x, y i z źródła, a także przyrostów pozycji, o które przesuniemy źródło. Po zainicjowaniu tych zmiennych ustalamy kilka właściwości źródła - mianowicie określamy, że chcemy, aby dźwięk źródłowy pętał, a następnie ustalamy jego początkową pozycję i prędkość. Właściwości prędkości są ważne dla efektu Dopplera. Jeśli zapomnisz ustawić właściwości prędkości, nie uzyskasz efektu Dopplera, nawet jeśli przesuniesz źródło, zmieniając jego współrzędne pozycji. Następnie źródło jest ustawione do odtwarzania i wprowadzana jest pętla, aby stale aktualizować położenie źródła co 100 milisekund. Kod w pętli po prostu dodaje przyrosty współrzędnych do bieżących współrzędnych źródła i sprawdza, czy źródło pozostaje w pewnych granicach. Jeśli źródło znajdzie się zbyt daleko, tłumienie będzie takie, że nie będziesz go już słyszeć, co po prostu staje się nudne. Pozostała część kodu zajmuje się sprzątnięciem po wyjściu. To naprawdę wszystko, aby tworzyć efekty dźwiękowe 3D za pomocą OpenAL. Oczywiście zarządzanie wieloma dźwiękami z efektami środowiskowymi w prawdziwej grze jest z pewnością bardziej zaangażowane, ale podstawy są takie same.