X. I wtedy cię uderza: potrzebujesz wykrywania kolizji

Bardzo często wierzysz, że znalazłeś rozwiązanie jakiegoś problemu, tylko po to, by zdać sobie sprawę, że zapomniałeś o czymś, co sprawia, że jest to tylko częściowe rozwiązanie. Czasami rozwiązanie, które znalazłeś, po prostu nie jest rozwiązaniem. Zrobiłem to bardzo. Czasami jesteś tak pewny rozwiązania, które zaczynasz implementować, tylko po to, aby uświadomić sobie, że tylko zmieniłeś problem w jeszcze jeden problem. Zwłaszcza w grafice, możesz często myśleć, że masz idealne rozwiązanie, kiedy to, co naprawdę masz, jest czymś w przybliżeniu z brzydkimi przypadkami, które tylko czekają na ujawnienie. Przynajmniej szukasz rozwiązań samodzielnie i nie zawsze odwołuję się do witryn takich jak http://www.mathworld.com lub podręczników. To pokazuje, że masz dobrą głowę na ramionach i potrafisz myśleć niezależnie. W części 4, "Podstawowe elementy geometryczne", poznałeś różne definicje niektórych podstawowych elementów geometrycznych. Nauczyłeś się nawet, jak znaleźć kolizję między dwoma elementami, a także metodę obliczania dokładnej lokalizacji kolizji. Ale czy to wszystko prawda? Czy naprawdę obliczyłeś dokładną lokalizację kolizji, czy też coś tam przegapiłeś? Trzeba pamiętać, że świat jest ogólnie dynamiczny. Wiele rzeczy na świecie ma tendencję do poruszania się, i tu właśnie niewiele pomaga rozdział 4. Zajmował się głównie statycznym światem i nie uwzględniał ruchu obiektu w czasie. Niestety, jeśli wystrzelisz rakietę w kosmos i po prostu sprawdzisz zderzenie rakiety z próbkami w czasie, możesz stracić potencjalne kolizje. Załóżmy na przykład, że wystrzeliwujesz rakietę na ścianie i z jakiegoś powodu szybkość klatek spada tak, że rakieta przesuwa się z jednej strony ściany na drugą w jednej ramce, nie dotykając samej ściany w żadnej z ramek. Oczywiście, jeśli spojrzysz na to z fizycznego punktu widzenia, rakieta powinna trafić w ścianę i eksplodować w tym punkcie. Ale ponieważ po prostu sprawdzałeś kluczowe klatki, nie wybrałeś tego. Oczywiście, to jest problem. W rzeczywistości dotyczy nawet najprostszych przypadków dynamicznego wykrywania kolizji. W bardziej złożonych przypadkach oba obiekty kończą się poruszaniem i musisz zweryfikować kolizję między dwoma ruchomymi obiektami. Tutaj przydaje się ten rozdział. To, czy chcesz wprowadzić czas jako zmienną w kolizji, to twój wybór. Ponownie, wszystko wraca do stworzenia lepszego modelu w porównaniu do prędkości. Podsumowując, tematy tej części są następujące:

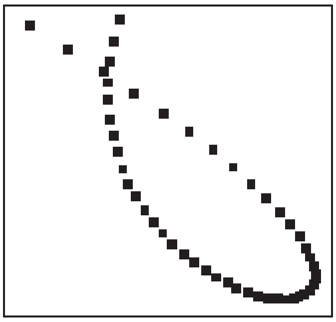
■ Zderzenia środowiskowe

■ Zderzenia obiektów między sobą

■ Wybieranie

Kolizje środowiskowe

Zacznijmy od łatwego spojrzenia na problem, w którym porusza się tylko jeden obiekt, a drugim obiektem jest płaszczyzna. Ze względów praktycznych zazwyczaj łatwiej jest podzielić świat na zestaw samolotów. Z tego powodu w tej sekcji widoczne będą tylko kolizje między płaszczyznami i innymi obiektami. Kiedy nadszedł czas, aby przenieść obiekt, matematycznie obiekt może poruszać się w dość skomplikowany sposób! Na przykład grawitacja może sprawić, że obiekt porusza się w sposób kwadratowy (to jest, krzywa wielomianowa drugiego stopnia). Jako inny przykład, cząstki tkaniny mogą poruszać się w bardzo złożony sposób, jak to opisuje ODE. W takim przypadku ruch jest niezwykle trudny do analizy, a wykrywanie kolizji może szybko ulec zepsuciu. Zamiast tego należy zastosować bardziej praktyczne podejście. Zamiast rozważać "prawdziwą trajektorię" obiektu, można po prostu przybliżać jego ruch, patrząc na trajektorię liniową obiektu wykonywanego między ramkami. Innymi słowy, przybliżasz ruch obiektu z klatki n - 1 do klatki n za pomocą linii prostej, jak pokazano na rysunku 10.1.



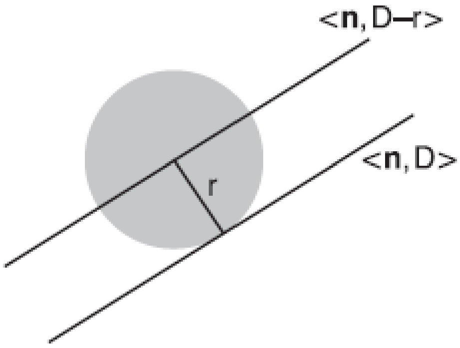
To znacznie upraszcza, ponieważ każdy ruch traktowany jest jako linia prosta. Linia jest również dość łatwą geometrią do pracy i nie powinna zbytnio komplikować równań. W niektórych przypadkach nawet prosty ruch liniowy może być dość trudny do pokonania, w porównaniu do przypadku, w którym nic się nie porusza. Spójrzmy na nasz pierwszy przypadek: kolizja kuli w czasie. Jeśli szukasz testu kolizji na linii prostej, zobacz Część 4, który opisuje kolizję odcinka linii trójkąta. Jest to w rzeczywistości bardziej precyzyjny test, ponieważ sprawdza również, czy kolizja znajduje się w obszarze trójkąta.

Kolizja z płaszczyznami kulistymi

W części 4 zobaczyłeś statyczną wersję tego scenariusza. Nauczyłeś się obliczać odległość między punktem w przestrzeni i płaszczyzną. W szczególności, równanie dla odległości d między płaszczyzną <n, D> i punktem p

$$d = D + \mathbf{n} \bullet \mathbf{p}$$

Teraz musisz dodać czynnik czasu do tego równania. Ponieważ mamy do czynienia ze sferą, odległość między kulą a płaszczyzną wynosi d - r, gdzie r jest promieniem kuli. Zilustrowano to na rysunku 10.2.



Jest to więc odpowiednik przesunięcia płaszczyzny o -r. Dlatego możesz spojrzeć na problem nie jako poruszającą się sferę, ale jako ruchomy punkt w przestrzeni. Równanie punktu poruszającego się wzdłuż linii jest następujące:

$$\mathbf{p}(t) = \mathbf{p}_0 + t(\mathbf{p}_0 - \mathbf{p}_1)$$

Jeśli połączysz oba równania (tj. Pozycję punktu do czasu, t = [0, 1]) i formułę odległości między płaszczyzną a punktem, otrzymasz równanie określające odległość między punktem a płaszczyzną w czasie:

$$d = D + \mathbf{n} \bullet \left(\mathbf{p}_0 + t \left(\mathbf{p}_0 - \mathbf{p}_1 \right) \right)$$

Kolizja występuje z płaszczyzną wtedy i tylko wtedy, gdy odległość od płaszczyzny i punktu wynosi 0. W ten sposób otrzymujesz

$$-D - \mathbf{n} \bullet \mathbf{p}_1 = t \cdot \mathbf{n} \bullet \left(\mathbf{p}_0 - \mathbf{p}_1 \right)$$

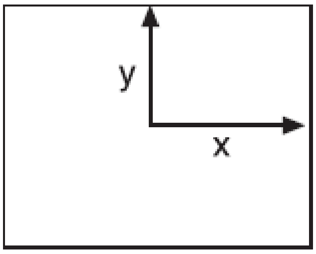
$$t = \frac{-D - \mathbf{n} \bullet \mathbf{p}_1}{\mathbf{n} \bullet \left(\mathbf{p}_0 - \mathbf{p}_1 \right)}$$

Na koniec, wyrażając równanie w taki sposób, aby uwzględniało promienie kuli, D '= (D - r), masz to:

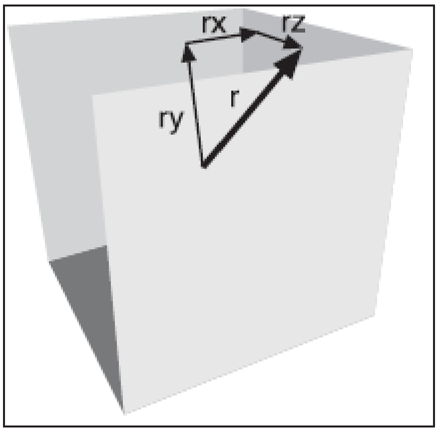
$$t = \frac{r - D - \mathbf{n} \bullet \mathbf{p}_1}{\mathbf{n} \bullet \left(\mathbf{p}_0 - \mathbf{p}_1 \right)}$$

Zderzenie z płaszczyzną pudła

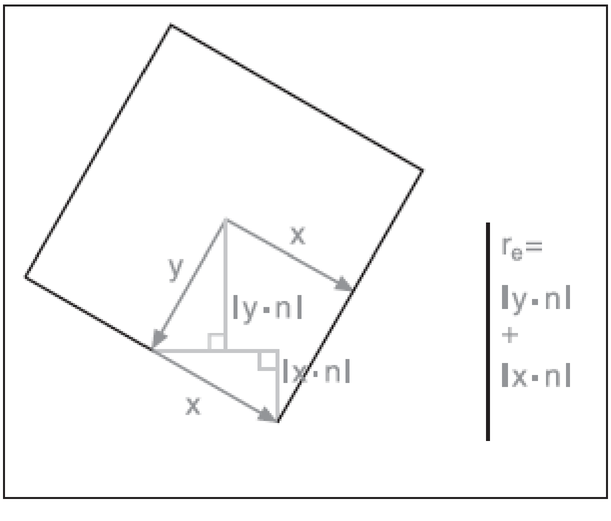
W ostatniej sekcji podejście było stosunkowo proste. Wziąłeś równanie opisujące minimalną odległość między punktem i płaszczyzną, a po prostu upewniłeś się, że promień obiektu był mniejszy niż odległość dostępna między punktem a płaszczyzną. A co, kiedy masz do czynienia z pudełkiem? Nie jest realistycznie myśleć o pudełku z promieniem. . . Albo to jest? Jeśli użyjesz standardu nieskończoności w polu, możesz rzeczywiście wykryć kolizje między pudłem a obiektem. Jedyny problem polega na tym, że nie pozwala to w ogóle na obrót pola, więc nie jest to zbyt użyteczne. Zamiast tego można uznać skrzynkę za zestaw trzech płaszczyzn, z których każda sprzężona jest z podwójnym odbiciem lustrzanym w punkcie środkowym pudła, jak pokazano na rysunku 10.3.



Oczywiście dotyczy to również osi z . Zatem wektory x, y i z pokazane na rysunku 10.2 rzeczywiście reprezentują przesunięcie pola od środka, a zatem nadają mu jego wymiar. Zauważ, że przesunięcie nie musi być wyrównane względem osi, więc ta notacja może reprezentować pole, które nie jest wyrównane względem osi. Jak to może pomóc? Sztuczka, której użyjesz, aby zaatakować ten problem, jest taka sama, jak w przypadku kuli. Kończysz obliczanie promienia pudełka. Może się to wydawać raczej głupim stwierdzeniem, ale promień nie jest tym, czego można się spodziewać. Nie jest to promień kuli, ale raczej długość pudełka od jego środka do płaszczyzny. Aby odróżnić ten promień od tego w sferze, nazwałbyś go skutecznym promieniem. Jeśli uda ci się z powodzeniem obliczyć efektywny promień dla pudełka, możesz zastosować tę samą technikę, którą widziałeś ze sferą, iw tym momencie wszystko stanie się zwykłymi szczegółami algebraicznymi. Rysunek 10.4 pokazuje ideę tego.



Jedno powinno być jasne w tym miejscu: Wektor efektywnego promienia, jak pokazano na rysunku 10.4, powinien mieć ten sam kierunek, co wektor normalny płaszczyzny. W rzeczywistości możesz spojrzeć na ten wektor jako minimalny wektor odległości między płaszczyzną a pudłem. Musi więc być prostopadła do płaszczyzny, a jej ogon musi być ustawiony w środku pudła. W tym miejscu potrzebna jest długość tego samego wektora. Trudne do wyliczenia, myślisz? Niezupełnie, jeśli rozumiesz, jak działają wektory. Wszystko sprowadza się do produktu dot. Zacznijmy od spojrzenia na problem w 2D. Zawsze łatwiej jest zwizualizować problem w niższych wymiarach przed pokonaniem wyższego końca. Można obliczyć efektywny promień dwuwymiarowego pudełka, znajdując miejsce, w którym linia rzucona ze środka pudełka na zewnątrz przecina się z czterema twarzami. Ale wymaga to dużej ilości obliczeń, nie wspominając o złożoności, którą dodaje. Alternatywnie możesz rozwiązać ten problem za pomocą produktu dot. Weź dwa wektory, które opisują pole, x i y. Jeśli spojrzysz na projekcję x na n, wektor efektywnego promienia (a także wektor normalny płaszczyzny), otrzymasz odległość x wzdłuż n. Zazwyczaj nie obejmuje to całego efektywnego promienia. Pozostała część jest kolejną projekcją na n, ale za pomocą wektora y, jak pokazano na rysunku 10.5.



Jeśli zsumujesz obie te długości, otrzymasz całkowitą długość efektywnego promienia. W 3D koncepcja jest jedynie rozszerzona. Jeśli istnieje dodatkowy wymiar, jest to tylko kwestia radzenia sobie z dodatkową projekcją dla wektora z na n. Wreszcie, musisz pamiętać, że podpisana jest projekcja. Innymi słowy, zależy to od kierunku, w którym zmierzasz. W ten sposób odległość może być dodatnia, ale może również być ujemna. Na przykład normalny wektor n może być rzutowany na jeden z wektorów, ale długość nie powinna się zmieniać, jeśli po prostu odzwierciedlasz kierunek. Aby to naprawić, wystarczy wprowadzić bezwzględną wartość do przewidywanej długości, aby upewnić się, że długości są traktowane bez względu na to, czy idą w kierunku wektorów pola, czy też x, yi z. Daje to następujące równanie dla efektywnego promienia re:

$$r_e = |\mathbf{x} \bullet \mathbf{n}| + |\mathbf{y} \bullet \mathbf{n}| + |\mathbf{z} \bullet \mathbf{n}|$$

Równanie do obliczenia czasu, w którym występuje przecięcie, nie różni się od równanie kuli, pod warunkiem że promień jest zmodyfikowany w celu odzwierciedlenia zmian. Jest to proponowane tutaj dla płaszczyzny <n, D> i ruchu od p_0 do p_1:

$$t = \frac{r_e - D - \mathbf{n} \bullet \mathbf{p}_0}{\mathbf{n} \bullet (\mathbf{p}_1 - \mathbf{p}_0)}$$

Oczywiście, obliczanie czasu kolizji nie jest bardzo skomplikowane. Co z obliczaniem położenia kolizji? Dla kuli jest to jedynie kwestia normalizacji normalnego wektora, tak aby jego długość pasowała do promienia kuli. Ale tutaj jest to nieco bardziej skomplikowane ze względu na nierównomierną geometrię obiektu. Najczęstszym przypadkiem jest sytuacja, gdy żaden z iloczynów skalarnych nie wynosi 0. Zwróć uwagę, że jedynym sposobem uzyskania iloczynu punktowego między n a jednym z wektorów definiujących pole, {x, y, z}, jest 0, jeśli wektor jest prostopadła do n. Jeśli żaden z wektorów nie jest prostopadły do n, pole uderzy w płaszczyznę przez jeden z jego wierzchołków narożnych. Wszystkie osiem rogów pola, w, są podane przez to równanie:

$$\mathbf{w} = \mathbf{p}(t) \pm \mathbf{x} \pm \mathbf{y} \pm \mathbf{z}$$

Wierzchami, które zderzają się z płaszczyzną, są te, których odległość od płaszczyzny jest równa lub mniejsza od odległości między płaszczyzną a innymi wierzchołkami. Innymi słowy, wierzchołki, które zderzają się z płaszczyzną, to te, które są najbliżej płaszczyzny. Minimalizacja odległości oznacza, że musisz zminimalizować produkt kropki:

$$\begin{aligned}\mathbf{w} \bullet \mathbf{n} &= (\mathbf{p}(t) \pm \mathbf{x} \pm \mathbf{y} \pm \mathbf{z}) \bullet \mathbf{n} \\ &= (\mathbf{p}(t) \bullet \mathbf{n}) \pm (\mathbf{x} \bullet \mathbf{n}) \pm (\mathbf{y} \bullet \mathbf{n}) \pm (\mathbf{z} \bullet \mathbf{n})\end{aligned}$$

Pierwszy termin jest stały, ponieważ t jest już określony i dlatego nie można go zminimalizować. Z drugiej strony możesz zminimalizować pozostałe warunki. Całkiem wyraźnie, terminy są minimalizowane, gdy cały termin, ±a • n, jest zminimalizowany. Dzieje się tak tylko wtedy, gdy każde z tych terminów jest negatywne. Tak więc, wszystko, co musisz zrobić, to wybrać znak taki, że wynika z niego najmniejsza wartość. Biorąc to pod uwagę, narożnik c, który przecina się z płaszczyzną, jest podany przez

$$\mathbf{c} = \mathbf{p}(t) + \operatorname{sgn}(\mathbf{x} \bullet \mathbf{n})\mathbf{x} + \operatorname{sgn}(\mathbf{y} \bullet \mathbf{n})\mathbf{y} + \operatorname{sgn}(\mathbf{z} \bullet \mathbf{n})\mathbf{z}$$

$$\operatorname{sgn}(i) = \left\{\begin{matrix} -1 & i > 0 \\ 1 & i < 0 \end{matrix}\right\}$$

Obserwuj, jak równanie nie jest faktycznie zdefiniowane dla iloczynu wektorowy równego 0. Gdy iloczyn ten wynosi 0, pole zderzy się z płaszczyzną z więcej niż jednym wierzchołkiem. W szczególności, jeśli jeden i tylko jeden produkt punktowy ma wartość 0, krawędź ramki zderza się z płaszczyzną.

Możesz znaleźć dwa rogi tej krawędzi, pozwalając, aby wybrany wierzchołek był dodatni lub ujemny. Innymi słowy, funkcja znaku, sgn, zmienia się na następujące:

$$\operatorname{sgn}(i) = \begin{Bmatrix} -1 & i > 0 \\ 1 & i < 0 \\ \pm 1 & i = 0 \end{Bmatrix}$$

Zwróć uwagę, jak ta definicja matematyczna działa również, gdy jedna z powierzchni pudła zderza się z płaszczyzną. W takim przypadku dwa produkty dotowane zostaną unieważnione, a oba znaki mogą być swobodnie pobierane, aby uzyskać zbiór wierzchołków, które tworzą zderzające się z płaszczyzną. Jedynym sposobem, w jaki można otrzymać trzy produkty z zerową kropką, jest skrzynka zdegenerowana. Jeśli myślisz o tym problemie geometrycznie, możesz zauważyć, że nie więcej niż cztery z ośmiu wierzchołków pudełka mogą kolidować z płaszczyzną, biorąc pod uwagę dowolny kąt dla pudełka i płaszczyzny.

Zderzenia obiektów między sobą

Wszystko to może stać się o wiele bardziej skomplikowane i wymagające dużego obciążenia procesora, gdy będziesz śledzić czas. Z tego powodu wiele gier ma tendencję do minimalizowania wykrywania kolizji międzyobiektowych. Większość czasu można zrobić bez niego, na przykład z iskrami w układzie cząstek. Z drugiej strony, prawdopodobnie chcesz mieć to wykrywanie, jeśli masz dwie postacie animowane, chyba że robisz grę z duchami. Podobnie do modelu kolizji obiekt-płaszczyzna dla środowiska, przybliżymy tylko ruch obiektów. Odbywa się to poprzez obserwację ruchu, jaki wykonali między poprzednią klatką a bieżącą ramką. Jest to ruch liniowy z jednego punktu do drugiego, który jest łatwiejszy w obsłudze. Po zapoznaniu się z równaniami z poprzedniej sekcji prawdopodobnie jeszcze bardziej docenisz to przybliżenie. Jeśli chcesz wprowadzić wielomian drugiego stopnia, potrzebujesz solwera kwadratowego zamiast solwera liniowego. Najwyraźniej nie chcesz wkładać rąk w tego rodzaju brudną wodę.

Kolizja Kuli i Kuli

Ah. . . jak zwykle, kula pojawia się jako pierwsza. Jest to rzeczywiście najłatwiejszy obiekt do pracy, więc jest całkiem dobrym kandydatem do analizy wykrywania kolizji pomiędzy dwoma obiektami w czasie. Pierwszy punkt jest dość prosty. Śledźmy losy sfer w czasie. Zdefiniuj następujące dwa równania, u, v, które definiują ruch każdej kuli w czasie, t:

$$\begin{aligned} \mathbf{u}(t) &= \mathbf{u}_1 + (\mathbf{u}_0 - \mathbf{u}_1)t \\ \mathbf{v}(t) &= \mathbf{v}_1 + (\mathbf{v}_0 - \mathbf{v}_1)t \end{aligned}$$

Możesz także łatwo zdefiniować odległość d pomiędzy tymi dwoma sferami w czasie:

$$\begin{aligned} d &= \left\| \mathbf{u}(t) - \mathbf{v}(t) \right\|_2 \\ &= \left\| \mathbf{u}_1 + (\mathbf{u}_0 - \mathbf{u}_1)t - \mathbf{v}_1 + (\mathbf{v}_0 - \mathbf{v}_1)t \right\|_2 \\ &= \left\| (\mathbf{u}_1 - \mathbf{v}_1) + (\mathbf{u}_0 - \mathbf{u}_1 + \mathbf{v}_0 - \mathbf{v}_1)t \right\|_2 \end{aligned}$$

W tym równaniu ukryty jest niechlujny pierwiastek kwadratowy, ale możesz się go pozbyć, jeśli po prostu obliczysz kwadrat odległości. Podczas porównywania odległości często inteligentnym posunięciem jest porównywanie kwadratu odległości. Nie ma to wpływu na faktyczne porównanie. Innymi słowy, porządek liczbowy ustalony przez równanie początkowe zostaje zachowany. Na szczęście upraszcza to twoją pracę. Aby nieco tu uprościć, pozwól mi zdefiniować dwie dodatkowe zmienne:

$$\mathbf{a} = (\mathbf{u}_1 - \mathbf{v}_1)$$
$$\mathbf{b} = (\mathbf{u}_0 - \mathbf{u}_1 + \mathbf{v}_0 - \mathbf{v}_1)$$

$$\begin{aligned} d^2 &= \|\mathbf{a} + \mathbf{b}t\|_2^2 \\ &= \mathbf{a} \bullet \mathbf{a} + 2\mathbf{a} \bullet \mathbf{b}t + \mathbf{b} \bullet \mathbf{b}t^2 \end{aligned}$$

Jedną z pierwszych rzeczy, które możesz chcieć zrobić, jest obliczenie istnienia kolizji. Jeśli w ogóle nie ma kolizji, nie ma sensu jej obliczać. Jest to łatwe do zrobienia dzięki znajomości rachunku różniczkowego. Musisz jedynie obliczyć pochodną i ustawić prędkość dystansu na 0. Ma to następujące konsekwencje:

$$\begin{aligned} d^2 &= \mathbf{a} \bullet \mathbf{a} + 2\mathbf{a} \bullet \mathbf{b}t + \mathbf{b} \bullet \mathbf{b}t^2 \\ d^2\,' &= 2\mathbf{a} \bullet \mathbf{b} + 2\mathbf{b} \bullet \mathbf{b}t \\ 0 &= 2\mathbf{a} \bullet \mathbf{b} + 2\mathbf{b} \bullet \mathbf{b}t \\ t &= -\frac{\mathbf{a} \bullet \mathbf{b}}{\mathbf{b} \bullet \mathbf{b}} \end{aligned}$$

Teraz chodzi tylko o podłączenie t do równania:

$$\begin{aligned} d^2 &= \mathbf{a} \bullet \mathbf{a} + 2(\mathbf{a} \bullet \mathbf{b})\left(-\frac{\mathbf{a} \bullet \mathbf{b}}{\mathbf{b} \bullet \mathbf{b}}\right) + (\mathbf{b} \bullet \mathbf{b})\left(-\frac{\mathbf{a} \bullet \mathbf{b}}{\mathbf{b} \bullet \mathbf{b}}\right)^2 \\ &= \mathbf{a} \bullet \mathbf{a} - \frac{2(\mathbf{a} \bullet \mathbf{b})^2}{\mathbf{b} \bullet \mathbf{b}} + \frac{(\mathbf{a} \bullet \mathbf{b})^2}{\mathbf{b} \bullet \mathbf{b}} \\ &= \mathbf{a} \bullet \mathbf{a} - \frac{(\mathbf{a} \bullet \mathbf{b})^2}{\mathbf{b} \bullet \mathbf{b}} \end{aligned}$$

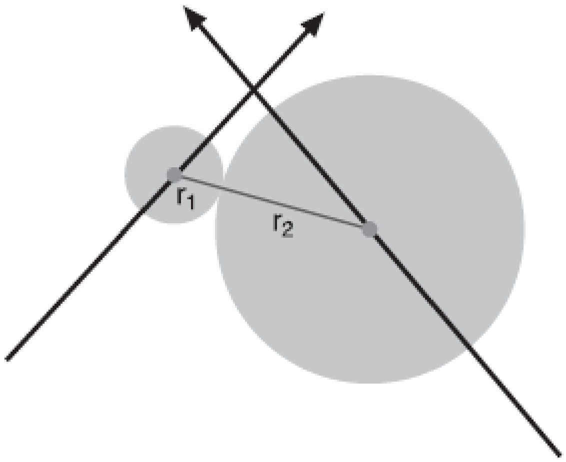
Ale golly, już wcześniej obliczyłeś to równanie, prawda? Tak, w części 4 nauczyłeś się obliczać minimalną odległość między dwiema liniami. To jest to samo równanie, z wyjątkiem tego, że jest kwadratem. Zaletą tego równania jest to, że porównanie jest szybsze, ponieważ nie trzeba obliczać pierwiastka kwadratowego wymaganego przez normalizację funkcji. Innym niezwykle ważnym aspektem jest to, że to równanie działa tylko dla nieskończonych linii. W tym przypadku mamy do czynienia z linią, która obejmuje dany zakres. Ważne jest, aby pamiętać, że czas obliczony w poprzednim równaniu nie jest czasem, w którym dwie sfery się przecinają. To tylko czas, w którym odległość między dwiema liniami jest minimalna. Jeśli kule skończą promień równy minimalnej odległości, czas będzie zgodny, ale rzadko tak jest. W związku z tym musisz obliczyć czas, w którym kule zderzają się, teraz, gdy wiesz, że mogą się przecinać. Teraz, kiedy faktycznie wiesz, że takie zderzenie może nastąpić, jeśli czas upłynie w nieskończoności, równanie kwadratowe przychodzi na ratunek, aby rozwiązać ten problem. Izolowanie dla wydajności t:

$$t = \frac{-2\mathbf{a}\bullet\mathbf{b} \pm \sqrt{(2\mathbf{a}\bullet\mathbf{b})^2 - 4(\mathbf{b}\bullet\mathbf{b})(\mathbf{a}\bullet\mathbf{a} - d^2)}}{2(\mathbf{b}\bullet\mathbf{b})}$$

$$= \frac{-2\mathbf{a}\bullet\mathbf{b} \pm 2\sqrt{(\mathbf{a}\bullet\mathbf{b})^2 - (\mathbf{b}\bullet\mathbf{b})(\mathbf{a}\bullet\mathbf{a} - d^2)}}{2(\mathbf{b}\bullet\mathbf{b})}$$

$$= \frac{-\mathbf{a}\bullet\mathbf{b} \pm \sqrt{(\mathbf{a}\bullet\mathbf{b})^2 - (\mathbf{b}\bullet\mathbf{b})(\mathbf{a}\bullet\mathbf{a} - d^2)}}{(\mathbf{b}\bullet\mathbf{b})}$$

W tym momencie masz dwie możliwe wartości. Geometrycznie to również jest prawdą. Jeśli istnieje skrzyżowanie, będą dwa punkty, w których dwie sfery będą miały jednakową odległość. Jesteś szczególnie zainteresowany pierwszą kolizją. Naprawdę nie dbasz o drugą, ponieważ pierwsza kolizja powinna w jakiś sposób odbić, a zatem druga kolizja (która tak naprawdę nie jest kolizją tak bardzo jak skrzyżowanie) nigdy nie nastąpi. Zatem najmniejsza wartość z poprzedniego równania pojawia się, jeśli wybierzesz ujemny pierwiastek kwadratowy. Ostatnią rzeczą, na którą musisz zwrócić uwagę, jest to, że d jest odległością pomiędzy centrami sfer. Dlatego musisz upewnić się, że centra sfer są sumą obu promieni, r_1 i r2, jak pokazano na rysunku 10.6.



W związku z tym czas zderzenia jest podany przez:

$$t = \frac{-\mathbf{a}\bullet\mathbf{b} \pm \sqrt{(\mathbf{a}\bullet\mathbf{b})^2 - (\mathbf{b}\bullet\mathbf{b})(\mathbf{a}\bullet\mathbf{a} - (r_1 + r_2)^2)}}{\mathbf{b}\bullet\mathbf{b}}$$

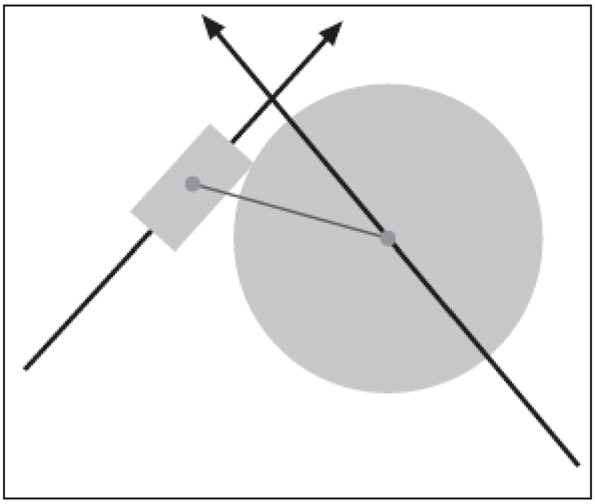
Przypomnijmy, że jedynym prawidłowym czasem, według definicji ruchu kuli, jest t = [0, 1]. Jeśli czas znajduje się poza tym zakresem, kolizja nie występuje, ponieważ zdarzenie miało miejsce w przeszłości (t <0) lub jeszcze nie nastąpiło (t> 1). Ewentualnie możesz przyciąć wartość t przy pierwszym teście, aby sprawdzić, czy kolizja w ogóle istnieje. Jeśli przyciąć wartość t w tym punkcie i obliczyć odległość za pomocą t, obliczymy minimalną odległość między dwiema liniami dla t = [0, 1]. W ten sposób możesz filtrować przypadek, w którym nie ma żadnych skrzyżowań. Oczywiście musisz tylko obliczyć czas zderzenia między dwiema sferami, gdy wiesz, że istnieje kolizja w twoim przedziale czasowym. Wreszcie, możesz być zainteresowany znalezieniem punktu kolizji. Może to być przydatne do odbijania kulek w przeciwnych kierunkach, do dodawania kalkomanii lub do celów odkształcania. Jest to całkiem proste, jeśli spojrzysz jeszcze raz na rysunek 10.6. Zauważ, że jest to wektor opisujący odległość między dwiema kulami w czasie t. W konsekwencji, jeśli chcesz znaleźć przecięcie drugiej sfery na pierwszej sferze, wystarczy upewnić się, że długość wektora odpowiada długości pierwszej sfery i tenże drugiej sfery. Oto równanie, które daje pozycję przecięcia $\{p_u = p_v\}$, w stosunku do pochodzenia świata:

$$\mathbf{p}_\mathbf{u} = \mathbf{u}(t) + r_\mathbf{u} \frac{\mathbf{v}(t) - \mathbf{u}(t)}{\left\| \mathbf{u}(t) - \mathbf{v}(t) \right\|_2}$$

$$\mathbf{p}_\mathbf{v} = \mathbf{v}(t) + r_\mathbf{v} \frac{\mathbf{u}(t) - \mathbf{v}(t)}{\left\| \mathbf{u}(t) - \mathbf{v}(t) \right\|_2}$$

Kolizja Kula-Pole

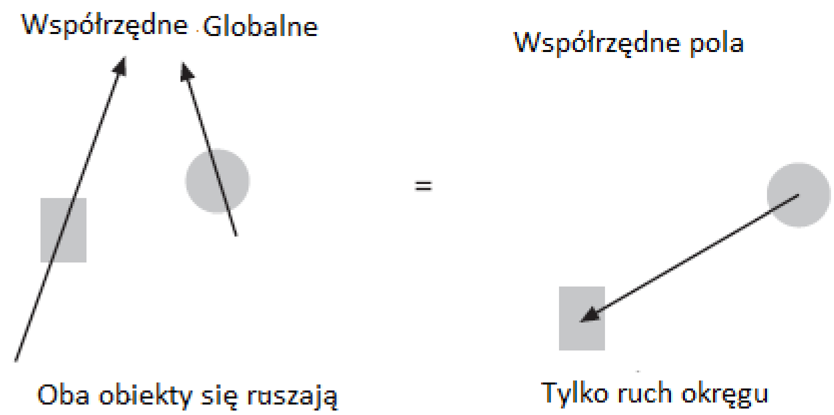
To wszystko staje się dla ciebie zbyt łatwe, prawda? W tych rozwiązaniach istnieje wyraźny wzorzec. Tu właśnie ten trend się niestety kończy. Pierwsze rozwiązanie, które przychodzi na myśl podczas obliczania kolizji sfery, jest prawdopodobnie podobne do wcześniejszych rozwiązań: Oblicz promień pola i sfery i pracuj z matematyką jak poprzednio. To jednak nie działa, ponieważ kolizja nie jest gwarantowana wzdłuż wektora między dwoma obiektami. Pokazano to na rysunku 10.7.



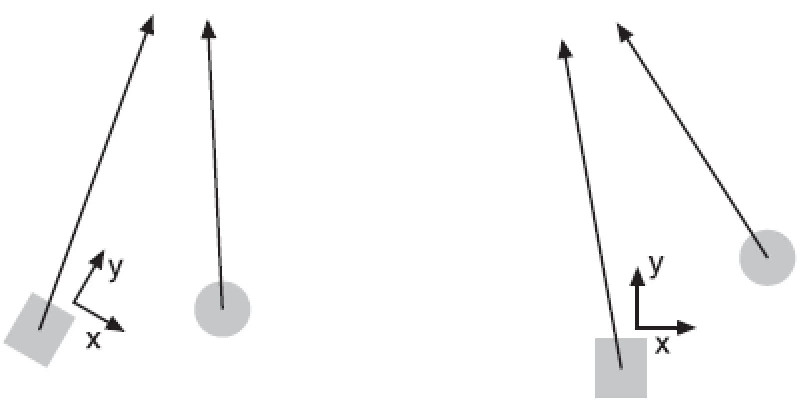
Oczywiście ten problem będzie wymagał trochę więcej przemyśleń. Jako szybki i brudny test, możesz zacząć od sprawdzenia, czy sfera i kula obejmujące to pole rzeczywiście przecinają się. To znacznie zmniejszy ilość obliczeń, które będziesz musiał wykonać. Odnalezienie promienia, r, sfery, która może zawierać ramkę zdefiniowaną przez trzy wektory przesunięcia, x, yiz, wymaga jedynie zastosowania twierdzenia Pitagorasa

$$r = \left\| \mathbf{x} + \mathbf{y} + \mathbf{z} \right\|_2$$

To częściowo rozwiązuje twój problem, ale tak naprawdę nie daje ci odpowiedzi, której szukasz. Niestety, nie ma eleganckiego rozwiązania tego problemu. Pierwszą sztuczką, którą powinieneś zastosować, jest zmiana referencji. Obliczając poprzednie przypadki, obliczyłeś wszystko w przestrzeni świata. Nic nie powstrzyma cię od obliczenia tego wszystkiego w innej przestrzeni. To nic nie zmienia dla istnienia kolizji. Kiedy nadszedł czas, aby obliczyć rzeczywistą kolizję, musisz zastosować odwrotność tej transformacji, aby uzyskać prawidłowy punkt kolizji. Szczególnie interesujące jest tu odniesienie do skrzynki. Pudełko jest tak naprawdę przyczyną problemów, ponieważ jego orientacja zmienia wszystko. Kula nie ma wpływu na orientację. Jeśli masz dwa poruszające się obiekty, każdy na linii prostej, możesz zmniejszyć problem do tylko jednego ruchomego obiektu, jeśli uważasz drugi obiekt za referencyjny, jak pokazano na rysunku 10.8.



Wystarczy odjąć wektor ruchu jednego obiektu od drugiego wektora. Niewyrównane pole również jest bolesne, ponieważ trzeba śledzić każdą oś. Z drugiej strony, jeśli pole jest wyrównane względem osi, jego płaszczyzny stają się równaniami postaci x = a, y = b, z = c, gdzie a, b i c są stałymi. Testy są łatwe do wykonania na tych płaszczyznach, ponieważ musisz martwić się tylko o jedną oś. Na przykład dla każdego punktu p, p - <a, 0, 0> dałoby odległość między punktem p a płaszczyzną x = a. Oczywiście jest to atrakcyjna opcja. Jak więc wykonać tę rotację? Z definicji twoje pudełko powinno być prostopadłe. Oznacza to, że wektory x, y, z, które opisują pole, tworzą podstawę w 3D. Jeśli myślisz o trzech wektorach jako osiach w świecie 3D, możesz przedefiniować podstawę kartezjańskiego układu współrzędnych, tak aby każdy wektor w tym nowym układzie współrzędnych był liniową kombinacją nowej podstawy. W skrócie można traktować wektory x, y i z jako nowy zestaw osi w nowej przestrzeni. Ostatnim wymogiem rotacji jest to, że transformacja musi być normalna. Dlatego musisz znormalizować każdy z twoich wektorów. Następnie można wygenerować macierz obrotu, która pobiera dowolny wektor w prostokątnej przestrzeni współrzędnych i przekształca ją w prostokątną przestrzeń współrzędnych, jak pokazano na rysunku 10.9.



Macierz, która to osiąga, jest trywialnie obliczana w następujący sposób (przy założeniu, że x, y i z są znormalizowane):

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} x_x & x_y & x_z \\ y_x & y_y & y_z \\ z_x & z_y & z_z \end{bmatrix}$$

Last but not least, może być łatwiej pracować z pudełkiem, jeśli jest wyśrodkowany na temat pochodzenia. Tak więc powinieneś najpierw zastosować tłumaczenie, tak aby pole było centrum wszechświata, a następnie obrócić tak, aby pole było wyrównane względem osi. Obliczanie wydajności złożonej matrycy

$$\mathbf{M} = \begin{bmatrix} x_x & x_y & x_z & -\mathbf{x}\bullet\mathbf{p} \\ y_x & y_y & y_z & -\mathbf{y}\bullet\mathbf{p} \\ z_x & z_y & z_z & -\mathbf{z}\bullet\mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Teraz możesz przetestować, że płaszczyzny boksu, {x, y, z}, mapują do podstawy, {<1, 0, 0>, <0, 1, 0>, <0,0, 1>}, i że środkowy punkt pola, p, mapuje do miejsca pochodzenia. Byłoby o wiele prostsze, gdybyś mógł trzymać pudełko w swojej statycznej pozycji, podczas gdy tylko kula poruszała się. Aby to osiągnąć, wystarczy spojrzeć na równanie ruchów każdego obiektu, u (t), v (t). Jeśli odejmiesz u (t), który jest zdefiniowany jako równanie ruchu skrzynki, z każdego z pozostałych równań ruchu, uzyskuje się następujące:

$$\mathbf{u}'(t) = \mathbf{u}(t) - \mathbf{u}(t) = 0$$

$$\begin{aligned} \mathbf{v}'(t) &= \mathbf{v}(t) - \mathbf{u}(t) = (\mathbf{v} - \mathbf{u})(t) \\ &= \mathbf{v}_1 - \mathbf{u}_1 + (\mathbf{v}_0 - \mathbf{v}_1 - \mathbf{u}_0 + \mathbf{u}_1)t \end{aligned}$$

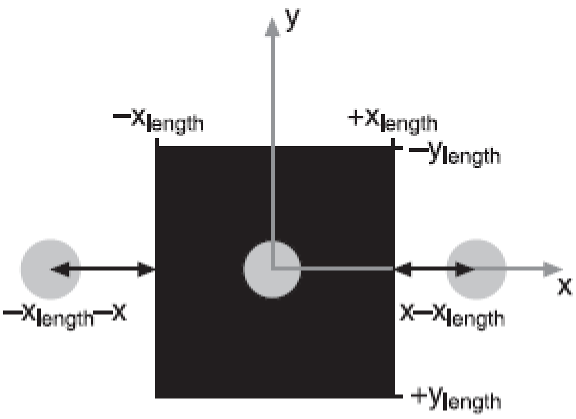
Możesz to zrobić, ponieważ interesuje cię tylko odległość między tymi dwoma obiektami, a odjęcie tej samej wartości dla każdego obiektu nie zmienia tej odległości. To jest to samo, co przydarzy się dwóm punktom w przestrzeni. Jeśli weźmiesz dwa punkty w przestrzeni i przesuniesz je o taką samą ilość w tym samym kierunku, odległość między tymi dwoma punktami się nie zmieni. Ta sama idea jest tutaj zastosowana, z wyjątkiem odejmowania funkcji zamiast odejmowania stałej. W rzeczywistości można sprawdzić, czy macierz transformacji nie zmienia odległości, obliczając wyznacznik macierzy (czyli podpisany obszar osi), który będzie równy 1 przez samą konstrukcję macierzy. Każdy obiekt, nad którym będziesz pracował w tej nowej przestrzeni współrzędnych, musi zostać przekształcony. Zatem macierz M musi przekształcić środek sfery:

$$\mathbf{v}''(t) = \mathbf{M}\mathbf{v}'(t)$$

Żeby było prostsze od tego miejsca, zdefiniuj ponownie v''(t) jako $q(t) = q_0 + \delta q$, a następnie określ długość pola jako {a, b, c}. Długość pola to oczywiście 2-norma wektorów {x, y, z}. W tym momencie istnieje dobrze znany algorytm, który można zastosować w całkowicie statycznym świecie. Algorytm ten zasadniczo podaje następujące informacje, które są stosowane dla każdej osi:

DistX = max (0, x - a, -a -x);

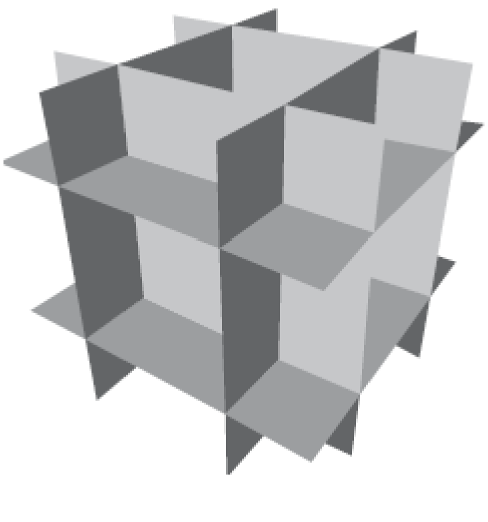
x jest zdefiniowany jako pierwszy składnik położenia sfery, podczas gdy a reprezentuje długość pudełka w x zdefiniowanym powyżej. To równanie określa odległość między punktem a polem tylko w x. Jeśli składnik x kuli znajduje się w pudełku, wynik wynosi 0. W przeciwnym razie odległość jest obliczana dla każdej płaszczyzny. Tutaj maksimum zapewnia, że wybrana zostanie prawidłowa płaszczyzna, a 0 w maksimum upewnia się, że gdy wartość x znajduje się pomiędzy dwoma płaszczyznami, odległość wynosi 0. Jest to zilustrowane na rysunku 10.10.



Możesz powtórzyć to dla osi Y i osi Z, aby ostatecznie obliczyć kwadrat odległości oddzielającej kulę od pudełka, po prostu obliczając normę tego wektora:

$$Dist^2 = DistX^2 + DistY^2 + DistZ^2$$

To nie było trudne, ale nadal nie rozwiązuje twojego problemu. To równanie nie zajmuje się czasem, a ze względu na wartość bezwzględną nie jest możliwe wyodrębnienie dla zmiennej t, jeśli zastosujemy tę samą sztuczkę, która była poprzednio użyta w przypadku wykrywania kolizji sfery i sfery. Tutaj muszą wybrać mniej eleganckie i ogólne rozwiązanie, dzieląc wartość bezwzględną na trzy różne równania. Jedynym problemem jest to, że ta bezwzględna wartość musi być zastosowana dla każdej współrzędnej. Oznacza to, że dostaniesz 3 x 3 x 3 przypadki, dając 27 różnych możliwości. W skrócie, pudełko dzieli się na 27 obszarów, jak pokazano na rysunku 10.11.



Ponieważ zakładasz, że początkowo nie ma skrzyżowania, musisz obliczyć tylko 26 równań. (Hej, co za ulga, myślałeś, że musisz obliczyć 27, prawda?) Więc jak wyglądają te równania? Zacznijmy od ustawienia notacji, aby ułatwić. Każda oś zostanie podzielona na trzy sekcje, <+1, 0,? 1>. W konsekwencji, jeśli środek kuli znajdował się w całkowicie dodatnim narożniku 27 obszarów, odnosiłbyś się do obszaru jako <+1, +1, +1>, który jest obszarem, w którym wartości składowe współrzędnych dla sfery p są większe niż odległość pudełka. Zacznijmy od tego jako przykładu.

Obliczanie przecięcia pola i sfery dla obszarów narożnych

W tym przypadku próbujesz obliczyć odległość między rogiem skrzynki a środkiem kuli. Równanie jest całkiem proste (i działa na wszystkie osiem zakrętów, o to chodzi). Dla sagi notacji zdefiniuj <a, b, c> jako l:

$$
\begin{aligned}
d^2 &= \left\| \mathbf{q}(t) - \langle a, b, c \rangle \right\|_2^2 \\
&= \left\| \mathbf{q}_0 + \Delta\mathbf{q}t - \mathbf{l} \right\|_2^2 \\
&= \left\| (\mathbf{q}_0 - \mathbf{l}) + \Delta\mathbf{q}t \right\|_2^2 \\
&= (\mathbf{q}_0 - \mathbf{l}) \bullet (\mathbf{q}_0 - \mathbf{l}) + 2(\mathbf{q}_0 - \mathbf{l}) \bullet \Delta\mathbf{q}t + \Delta\mathbf{q} \bullet \Delta\mathbf{q}t^2
\end{aligned}
$$

Czy to nie jest znajomy pierścień? W tym momencie kontynuujesz izolację t:

$$
\begin{aligned}
t &= \frac{-2(\mathbf{q}_0 - \mathbf{l}) \bullet \Delta\mathbf{q} \pm \sqrt{\left(2(\mathbf{q}_0 - \mathbf{l}) \bullet \Delta\mathbf{q}\right)^2 - 4(\Delta\mathbf{q} \bullet \Delta\mathbf{q})\left((\mathbf{q}_0 - \mathbf{l}) \bullet (\mathbf{q}_0 - \mathbf{l})\right)}}{2\Delta\mathbf{q} \bullet \Delta\mathbf{q}} \\
&= \frac{-(\mathbf{q}_0 - \mathbf{l}) \bullet \Delta\mathbf{q} - \sqrt{\left((\mathbf{q}_0 - \mathbf{l}) \bullet \Delta\mathbf{q}\right)^2 - (\Delta\mathbf{q} \bullet \Delta\mathbf{q})\left((\mathbf{q}_0 - \mathbf{l}) \bullet (\mathbf{q}_0 - \mathbf{l})\right)}}{\Delta\mathbf{q} \bullet \Delta\mathbf{q}}
\end{aligned}
$$

Oczywiście tutaj nie jest możliwa optymalizacja. Ponownie bierzesz najmniejszy root i odejmujesz pierwiastek kwadratowy z tych samych powodów, o których wspominałeś wcześniej. Jeśli pierwiastek kwadratowy jest ujemny, nie ma przecięcia. Co więcej, aby skrzyżowanie w ogóle istniało, czas musi znajdować się w jego zasięgu. Na przykład, jeśli sfera pozostaje w tym obszarze przez cały czas jej trwania, można oczekiwać, że zakres będzie wynosił [0, 1]. Jeśli sfera zmienia obszar pośrodku, musisz obliczyć nowy przedział czasowy. Wkrótce zobaczysz to na większej głębokości. Jeśli przecięcie rzeczywiście miało miejsce, wystarczy obliczyć wektor l - q (t). To da ci wektor, który przechodzi od środka kuli do pudełka. Przypomnijmy, że ten wektor znajduje się teraz w polu przestrzeni. Dlatego musisz przekonwertować przesunięcie wektora z przestrzeni pudełkowej na przestrzeń światową. W tym miejscu przydatna jest matryca odwrotna. Musisz obliczyć odwrotną macierz M i pomnożyć ją przez wektor korekcji. Na szczęście, ponieważ M jest ortonormalne, odwrotnością M naprawdę jest M^T. Tak więc, światowa pozycja kolizji c jest

$$
\mathbf{c} = \mathbf{v}(t) + r\mathbf{M}^{\mathrm{T}}\left(\mathbf{l} - \mathbf{q}(t)\right)
$$

Oczywiście ta strategia będzie działać dla każdego rogu pudełka. Wystarczy zmienić znaki komponentów.

Obliczanie przecięcia pola i sfery dla jednego lub dwóch krawędzi

Co się dzieje, gdy nie masz rogu? Co zaskakujące, sprawy zaczynają być nieco łatwiejsze. Znacznie trudniej jest wyrazić wszystko w kategoriach wektorów, ale proces jest prawie taki sam. Jedyna różnica polega na tym, że jedna z odległości zostanie ustawiona na 0. Na przykład obliczenie czasu dla obszaru <+1, +1, 0> oznaczałoby, że należy anulować trzecią zmienną. Innymi słowy, obliczenia są następujące:

$$
\begin{aligned}
d^2 &= \left\| \left(\mathbf{q}(t) - \mathbf{l} \right) \& \left(1,1,0\right) \right\|_2^2 \\
&= \left\| \left\langle \mathbf{q}_x(t), \mathbf{q}_y(t) \right\rangle - \left\langle a, b \right\rangle \right\|_2^2
\end{aligned}
$$

Długość jest w pełni opisana tylko przez x i y, jak opisano w równaniu maksymalnym podanym wcześniej. Zasadniczo pochodzi to samo równanie. Jedyna różnica polega na tym, że całkowicie pomijasz składnik z. Wreszcie, możesz zainteresować się obliczaniem rzeczywistej pozycji kolizji. W tym przypadku wystarczy pamiętać o zmiennej, o której zapomniałeś, i obliczyć różnicę między zasięgiem pola, l, a pozycją kuli, gdy się koliduje, q(t). Jeśli chodzi o zmienną, która została zamaskowana z równania, musi pozostać jako 0. Tak więc, dla poprzedniego przykładu, wektor korekcji z kuli zostanie podany przez rzeczy są bardzo podobne, gdy masz przecięcie z płaszczyzną.

$$
\begin{aligned}
\mathbf{c} &= \mathbf{v}(t) + r\mathbf{M}^{\mathrm{T}} \left(\left(\mathbf{l} - \mathbf{q}(t) \right) \& \left\langle 1,1,0 \right\rangle \right) \\
&= \mathbf{v}(t) + r\mathbf{M}^{\mathrm{T}} \left\langle \mathbf{l}_x - \mathbf{q}_x(t), \mathbf{l}_y - \mathbf{q}_y(t), 0 \right\rangle
\end{aligned}
$$

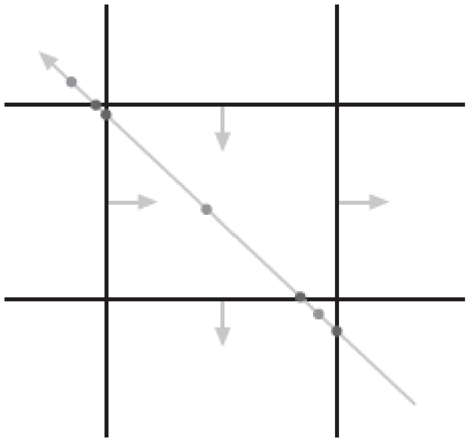
W takim przypadku twoje dwa składniki są zamaskowane, a tylko jeden składnik nie jest. W związku z tym obliczenia stają się dość proste, ponieważ trzeba zamaskować dwie zmienne i zachować tylko jedną. Tak więc wektor generowany przez ten typ przecięcia jest zawsze prostopadły do płaszczyzny (jest to normalna płaszczyzna z długością kuli).

Znając ten obszar

Teraz już wiesz, jak znaleźć kolizję, ale nadal nie wiesz, jak określić długość 27 obszarów. To całkiem proste. Wystarczy sprawdzić, czy każdy składnik jest mniejszy niż określona wartość, większa niż określona wartość lub pośrednia. To drastycznie upraszcza i przyspiesza kod. Pierwszym testem, który powinieneś wykonać, jest sprawdzenie, czy oba punkty końcowe ruchu kuli znajdują się w tym samym obszarze. Jeśli tak, staje się całkiem jasne i łatwe. Czas, t, ma zakres [0, 1], a ty masz tylko jedną kontrolę do zastosowania. Jeśli tak nie jest, staje się nieco obrzydliwe. Dla ogólnego algorytmu, możesz przejść przez wszystkie sześć płaszczyzn i obliczyć czas, w którym linia przecina się z płaszczyzną (część 4 zawiera wszystkie szczegóły wymagane do tego). Kiedy już znasz czasy dla wszystkich sześciu samolotów, musisz wybrać minimalny czas. Każda wartość mniejsza niż 0 lub większa niż 1 powinna zostać odrzucona natychmiast, ponieważ oznacza to, że skrzyżowanie nigdy nie zostanie osiągnięte w tym segmencie. Jeśli zachowasz posortowaną listę czasów z przypisanymi płaszczyznami, możesz określić kolejność, w której będziesz przemierzać płaszczyzny w czasie t = [0, 1]. Po ustaleniu pierwszego przecięcia z płaszczyzną obszaru, należy najpierw sprawdzić, czy istnieje przecięcie między kulą a ramką między czasem t = [0, i], gdzie reprezentuję czas obliczony na skrzyżowanie między płaszczyznami obszaru. Najtrudniejszą częścią tego jest określenie, w którym obszarze się poruszasz, biorąc pod uwagę, że znalazłeś skrzyżowanie i płaszczyznę. Jak się okazuje, jest naprawdę łatwy sposób na zrobienie tego. Jeśli umieścisz początkowe centrum sfery w równaniu płaszczyzny, możesz określić, czy wierzchołek znajduje się powyżej, czy pod płaszczyzną, jak już wcześniej widziałeś:

$$\mathbf{n} \bullet \mathbf{p} + D = s$$

Tutaj s jest znakiem położenia punktu, p, w odniesieniu do płaszczyzny, <n, D>. Jeśli wynik s jest dodatni, oznacza to, że znajdujesz się w tym samym półpłaszczyźnie, co wskazywał na normalny wektor samolotu. Odwrotnie, jeśli wynik jest ujemny, jesteś w drugim pół-planie. W związku z tym musisz tylko stworzyć konwencję z tego. Na przykład, powiedz, że górny lewy kierunek to miejsce, w którym normalne punkty płaszczyzny będą wskazywać podczas obliczania przecięcia. W konsekwencji, jeśli środek sfery był początkowo w pozytywnym regionie (co sugeruje to samo, co wskazywał normalny wektor) i kończy się ujemnym po wybraniu punktu po czasie kolizji, oznacza to, że się poruszyłeś jeden obszar po prawej w części samolotu. I odwrotnie, jeśli przejdziesz z pozytywnej na negatywną, przesunąłeś jeden obszar w prawo. Rysunek 10.12 pokazuje ten pomysł w 2D.



W 3D można pomyśleć o tych płaszczyznach jako o siatce 3 x 3 (powiedzmy środkowej). Tutaj równania płaszczyzny byłyby podobne do {<1, 0, 0, || x ||>, <1, 0, 0, - || x ||>, <0, 1, 0, || y ||>, <0, 1, 0, - || y ||>}. Środek kuli znajduje się w dolnym prawym obszarze, który przyjmujesz jako wartości dodatnie. Obliczenie tego równania z centrum dałoby wartość s = 1 dla każdej płaszczyzny wspomnianej powyżej. Po wykryciu pierwszego przecięcia z płaszczyzną, możesz obliczyć średni punkt między rozpiętością drugiego obszaru, aby uzyskać punkt znajdujący się w nowym obszarze, i umieścić ten punkt w poprzednim równaniu, aby uzyskać wartość ujemną. Ponieważ przechodzisz z dodatniego na ujemny, wywnioskujesz, że kierunek na x jest przesunięcie o jeden obszar powyżej. Podobnie, następne przecięcie jest w płaszczyźnie siedzącej w y, i przechodzi od dodatniej do ujemnej, co oznacza, że wartość x powinna być przesunięta o jedną jednostkę w lewo. Gdyby stało się coś przeciwnego, musiałbyś przesunąć jeden obszar w przeciwnym kierunku. Jeśli śledzisz obszar za pomocą wektora obszaru z potencjalnymi wartościami {-1, 0, +1}, jest to bardzo proste, ponieważ wystarczy dodać 1 do lub odjąć 1 od odpowiedniej płaszczyzny komponentu. Oczywiście, nie jest to łatwy problem, który widzieliście wcześniej w tej części, a kilka szybszych, ale mniej precyzyjnych testów powinno być przeprowadzonych przed tym, aby zminimalizować obliczenia. Powinieneś ocenić, czy naprawdę musisz to zrobić, lub czy szybsze przybliżenie jest warte twojego czasu.

Zderzenie Pole-Pole

Widząc, jak brzydki był ten ostatni przypadek, prawdopodobnie nie możesz się doczekać, aby przeczytać ten. Znalazłeś już test, aby sprawdzić, czy skrzynka przecina się z płaszczyzną. Możesz użyć tego testu, aby sprawdzić, czy skrzynki się kolidują. Podobnie jak w przypadku wykrywania kolizji sfery, można zdefiniować wektor przesunięcia pola A jako {x, y, z}. Jeśli przekształcisz przestrzeń współrzędnych w przestrzeń współrzędną pola A, otrzymasz tę samą macierz transformacji, jaką obliczałeś poprzednio:

$$\mathbf{M} = \begin{bmatrix} x_x & x_y & x_z & -\mathbf{x} \bullet \mathbf{p} \\ y_x & y_y & y_z & -\mathbf{y} \bullet \mathbf{p} \\ z_x & z_y & z_z & -\mathbf{z} \bullet \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

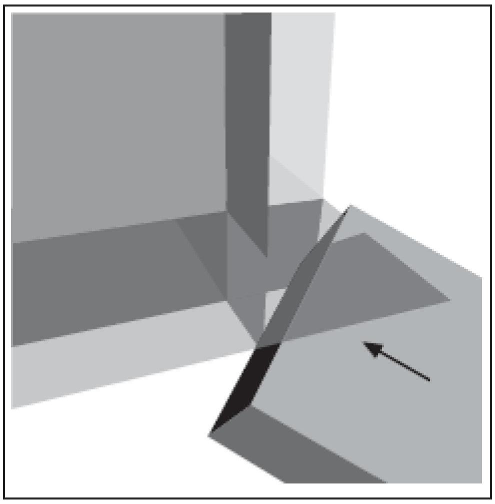
Pierwszą rzeczą, na którą powinieneś zwrócić uwagę, jest to, że musisz sprawdzić tylko trzy płaszczyzny w stosunku do pudełka, a nie cały zestaw sześciu samolotów. Krótko mówiąc, jeśli weźmiesz dowolne pudełko i obrócisz je w dowolnym kierunku, nie zobaczysz więcej niż trzech jego twarzy. Tylko dla celów tego problemu, biorąc pod uwagę normalny dla płaszczyzny n i wektor, który jest pomiędzy p_a i p_b, możesz obliczyć widoczność twarzy obliczając produkt kropki. Znak będzie sygnalizował, czy twarz <n, D> jest widoczna z punktu widzenia p_b patrząc w kierunku p_a.

$$\mathbf{n} \bullet (\mathbf{p}_A - \mathbf{p}_B) = s$$
$$\begin{Bmatrix} \text{Widoczne} & s > 0 \\ \text{Nie Widoczne} & s <= 0 \end{Bmatrix}$$

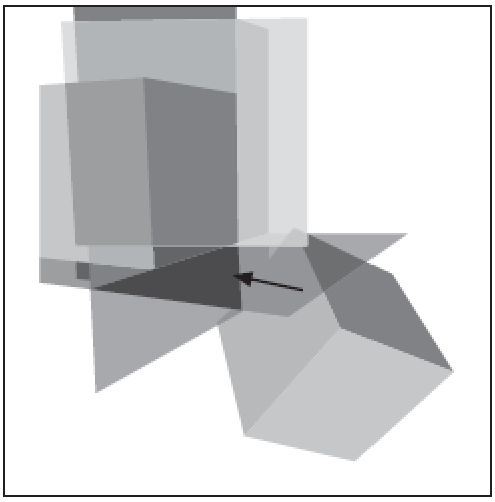
Po znalezieniu trzech widocznych płaszczyzn możesz sprawdzić je, sprawdzając, czy nie ma kolizji z polem B. Jeśli zauważysz, że kolizja istnieje, nie oznacza to, że kolizja między tymi dwoma skrzyniami rzeczywiście nastąpiła. Najpierw musisz sprawdzić, czy przecięcie płaszczyzny i pola B rzeczywiście znajduje się w obrębie pola A. Płaszczyzna rozciąga się do nieskończoności, podczas gdy twarz jest skończona w przestrzeni. W związku z tym należy sprawdzić, czy punkt przecięcia znajduje się w granicach odpowiadającej mu ściany, stosując następującą weryfikację:

$$-\|\mathbf{x}\|_2 \le \mathbf{c}_x \le \|\mathbf{x}\|_2$$
$$-\|\mathbf{y}\|_2 \le \mathbf{c}_y \le \|\mathbf{y}\|_2$$
$$-\|\mathbf{z}\|_2 \le \mathbf{c}_z \le \|\mathbf{z}\|_2$$

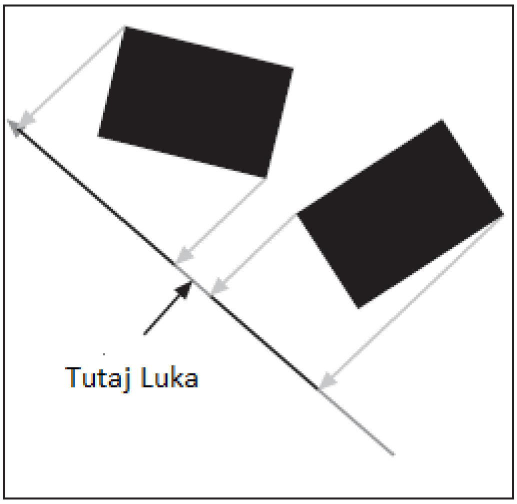
Oczywiście jeden z testów będzie bez znaczenia. Kolizja występuje z tym samym płaszczyzną, więc wiesz, że minie. Dlatego tylko dwa z tych testów są wymagane na twarz. Jeśli testowałeś przeciwko twarzy x, wiesz, że x będzie w granicach, a zatem wystarczy przetestować dla y i z. Jeśli znajdziesz kolizję w granicach, kolizja jest pierwszą kolizją między skrzyniami, a następnie możesz wyjść. Ten test wydaje się dość prosty. Czy obejmuje wszystkie możliwe kolizje? Innymi słowy, czy możesz stwierdzić, że we wszystkich innych przypadkach dwa pola nie kolidują? Oczywiście, że nie, głupie, to byłoby zbyt łatwe. Na przykład powiedzmy, że pole B trafia do pola A wewnątrz powierzchni poprzez jeden z jej wierzchołków narożnych. W takim przypadku można łatwo znaleźć kolizję między dwoma skrzyniami w pierwszym kroku opisanym powyżej. Załóżmy jednak, że kolejność pudełek jest odwrócona. Innymi słowy, wybrałeś pole B jako pole statyczne, a pole A jako ruchome pole. W tym przypadku pole A zbliża się do pola B od kąta narożnika i od płaskiego kierunku, jak pokazano na rysunku 10.13.



Problem z tym przypadkiem polega na tym, że rzeczywiście znajdziesz trzy zderzenia między dwoma pudłami (po jednym na samolot), ale w przypadku każdego samolotu kolizja, którą znajdziesz, będzie poza zasięgiem i nie przejdzie testu zderzenia, ponieważ pokazano na rysunku 10.13. Ale na szczęście, ponieważ odwracające się pola A i B działają, możesz dodać kolejny zestaw trzech testów, testując pole B na polu A. Aby podsumować, musisz przetestować trzy pola A w polu B, a także musisz przetestować trzy płaszczyzny B w stosunku do pola B. Które zajmują się tym przypadkiem, ale czy obejmuje on wszystkie możliwe kolizje? Co jeśli dwa pola przecinają się ze sobą na krawędzi i żadna z ich osi nie jest równoważna, jak pokazano na rysunku 10.14?



W takim przypadku wszystkie sześć testów przeciwko samolotom nie znajdzie prawidłowego punktu przecięcia. Testy, które wykryją kolizję, nie wykryją przecięcia między dwoma polami, ponieważ przecięcie nie będzie znajdować się w granicach płaszczyzny. Houston, mamy problem! Oczywiście ta metoda ma wady. Może wykryć zderzenia dwóch skrzynek w większości przypadków, ale nie w każdym przypadku. Możesz podejść do tego problemu pod podobnym kątem do poprzedniej sekcji, ale zamiast tego spojrzysz na niego pod zupełnie innym kątem. Pierwsza sekcja (przy użyciu testu przecięcia płaszczyzn płaszczyzny) używa twierdzenia zwanego twierdzeniem płaszczyzny oddzielającej. To bardzo oczywiste twierdzenie stwierdza, że jeśli można znaleźć płaszczyznę w przestrzeni, która oddziela dwie geometrie tak, że każdy punkt jednej geometrii znajduje się po dodatniej stronie płaszczyzny, podczas gdy punkty innej geometrii znajdują się po ujemnej stronie płaszczyzny , dwa obiekty nie kolidują. Jest to dość oczywiste. Geometrycznie, gdy oba obiekty zderzają się, oznacza to, że nie ma płaszczyzny, która może oddzielić dwa obiekty. Działa to całkiem nieźle, gdy umiesz pozycjonować samolot w kosmosie i działa idealnie w testach płaskich 2 x 3, które wykonałeś początkowo. Lemat tego testu jest twierdzeniem o osi rozdzielającej. Twierdzenie to stwierdza, że jeśli istnieje wektor kierunkowy l, dla którego projekcja kształtów na tym wektorze nie przecina się, kształty nie przecinają się. Twierdzenie to wykorzystuje tę samą ideę, co inne twierdzenie. Jeśli pomiędzy tymi dwoma obiektami istnieje luka, rzutując obiekty na linii (tak jak można zrobić ortogonalnie dla cienia), projekcja również się nie przecina. Idea tego twierdzenia została przedstawiona na rysunku 10.15.



Jeśli chcesz, możesz wybrać trzy normalne wektory pól jako kierunkowe. Jeśli obliczysz rzut kształtów na te wektory, uzyskasz taki sam rezultat, jak w teście przecięcia płaskiej skrzyni. Test przecięcia płaskich skrzynek jest trochę bardziej pomocny, w tym sensie, że zapewnia on bezpośrednio czas przecięcia i pozycję kolizji. Projekcja obiektów na linii może wydawać się trudnym zadaniem, ale tak naprawdę nie jest. W rzeczywistości nauczyłeś się już projektować pudełko na wektorze. Oto równanie, które to robi:

$$r_e = |\mathbf{x} \bullet \mathbf{l}| + |\mathbf{y} \bullet \mathbf{l}| + |\mathbf{z} \bullet \mathbf{l}|$$

Wygląda jak znajome równanie? Powinno być. Testowanie projektowanego skrzyżowania jest dość proste. Jeśli możesz sprawdzić, czy suma promieni pola jest mniejsza niż odległość rzutowana między środkami dwóch pudełek, udowodniłeś, że istnieje różnica między tymi dwoma obiektami i nie kolidują one ze sobą. Matematycznie, to, co próbujesz znaleźć, jest następujące dla pól centrum u i v, z odpowiednimi promieniami r_{ea} i r_{ec} oraz wektorem rzutowania l:

$$\mathbf{l} \bullet (\mathbf{u} - \mathbf{v}) \geq r_{ea} + r_{rb}$$

Kiedy okaże się, że ten test nie jest prawdziwy, wtedy musisz spróbować znaleźć inny odcinek l, który zaspokoi tę nierówność. Nieodnalezienie takiego wektora oznacza, że obiekty przecinają się. Aby być decydującym w tym teście, musisz znaleźć pełny zestaw wektorów projekcyjnych, l, które umożliwią ci sprawdzenie każdego możliwego przypadku skrzyżowań skrzynek. Jedynym przypadkiem pominiętym przez test wykrywania kolizji w płaszczyźnie jest sytuacja, gdy dwie krawędzie przecinają się. Co ciekawe, jeśli zbudujesz płaszczyznę, która była normalna dla obu krawędzi, możesz określić, czy oba obiekty przecinały ten punkt, czy nie. Jedynym problemem jest znalezienie pozycji samolotu. Jeśli zamiast tego użyjesz testu twierdzenia osi, nie musisz umieszczać niczego w przestrzeni, ponieważ projekcja jest wykonywana na wektorze kierunkowym, a nie zakotwiczonym wektorem pozycji w przestrzeni. To jest piękno tego testu. Testy mają także niezwykle ścisły związek. Ze względu na ortogonalność skrzynek, norma płaszczyzny, która jest sprawdzana dla pierwszego testu, może być użyta jako wektor kierunkowy dla innej pary krawędzi. Jeśli myślisz w kategoriach krawędzi, każde pudełko jest wyposażone w trzy zestawy krawędzi. Każda krawędź z jednego pudełka może przecinać dowolną krawędź z drugiej. Oznacza to, że musisz obliczyć dodatkowe dziewięć krawędzi, które są niczym innym jak każdym możliwym poprzecznym produktem między krawędziami dwóch pudełek. Jeśli pudełka nie były prostopadłe (a więc nie były w ogóle pudełkami, ale czworokąty), musielibyśmy dodać produkt krzyżowy między każdą możliwą parą niezależnych krawędzi. Podsumowując, twierdzenie o osi rozdzielającej musi być zweryfikowane w sumie 15 wektorów kierunkowych, jeśli wykonywane są niezależnie (czyli bez twierdzenia o płaszczyźnie separującej). Te samoloty to sześć normalnych płaszczyzn z dwóch skrzynek i unikalny iloczyn krzyżowy każdej możliwości między tymi sześcioma normalnymi. Matematycznie:

$$\mathbf{l} = \left\{ \begin{matrix} \mathbf{x}_a, \mathbf{y}_a, \mathbf{z}_a, \\ \mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b, \\ \mathbf{x}_a \times \mathbf{x}_b, \mathbf{x}_a \times \mathbf{y}_b, \mathbf{x}_a \times \mathbf{z}_b, \\ \mathbf{y}_a \times \mathbf{x}_b, \mathbf{y}_a \times \mathbf{y}_b, \mathbf{y}_a \times \mathbf{z}_b, \\ \mathbf{z}_a \times \mathbf{x}_b, \mathbf{z}_a \times \mathbf{y}_b, \mathbf{z}_a \times \mathbf{z}_b \end{matrix} \right\}$$

Ten zestaw to kompletny zestaw wymagany dla statycznych skrzynek w przestrzeni. Tak, słowo kluczowe tutaj jest statyczne. Wszystko to wciąż nie bierze pod uwagę czasu. Biorąc pod uwagę twój test braku przecięcia, nie jest bardzo skomplikowane dodawanie do niego pojęcia czasu. Możesz po prostu pozwolić u i v być funkcją czasu i zastosować test do zakresu t = [0, 1]:

$$\mathbf{l} \bullet \left(\mathbf{u}(t) - \mathbf{v}(t) \right) \geq r_{ea} + r_{rb}$$

Ponieważ ruch (u (t) - v (t)) jest liniowy, po prostu trzeba pokazać, że nie ma przecięcie na początku lub na końcu, a kierunek rzutu dla początek i koniec nie zmieniły znaku. Na przykład, jeśli pudełko zaczyna się po prawej stronie drugiego pola, a na końcu przesuwa się na lewo od drugiego pola bez przecinania w żadnym punkcie, nie jest to kompleksowy test. Fakt, że pudełka powinny pozostawać w tych samych względnych pozycjach względem siebie, rozwiązuje ten problem. Matematycznie kontrola sprowadza się do:

$$\begin{gathered} \mathbf{l} \bullet \left(\mathbf{u}(0) - \mathbf{v}(0) \right) \geq r_{ea} + r_{rb} \\ \mathbf{l} \bullet \left(\mathbf{u}(1) - \mathbf{v}(1) \right) \geq r_{ea} + r_{rb} \\ \operatorname{sgn}\left(\mathbf{l} \bullet \left(\mathbf{u}(0) - \mathbf{v}(0) \right) \right) = \operatorname{sgn}\left(\mathbf{l} \bullet \left(\mathbf{u}(1) - \mathbf{v}(1) \right) \right) \end{gathered}$$

Ten test należy zastosować do 15 poprzednich samolotów. Jeśli którykolwiek z testów minie, oznacza to, że pudełka faktycznie się nie kolidują. Ruch trochę tu komplikuje. W rzeczywistości test 15-płaszczyznowy nie jest wystarczająco kompleksowy, aby wystąpić kolizja, jeśli wszystkie testy zakończą się niepowodzeniem. Tak, musisz dodać jeszcze więcej kierunkowych wektorów. Przypadek, który nie jest objęty ruchem w czasie, ma miejsce, gdy wektor kierunkowy jest tak dobrany, aby był prostopadły do ruchu między dwoma przedmiotami. Aby uprościć zapis, możesz zdefiniować wektor w (t) jako wektor u (t) - v (t). A zatem:

$$\begin{aligned}\mathbf{u}(t)-\mathbf{v}(t) &= \mathbf{u}_1+(\mathbf{u}_0-\mathbf{u}_1)t-(\mathbf{v}_1+(\mathbf{v}_0-\mathbf{v}_1)t) \\ &= (\mathbf{u}_1-\mathbf{v}_1)+(\mathbf{u}_0-\mathbf{u}_1-\mathbf{v}_0+\mathbf{v}_1)t \\ \mathbf{w}(t) &= \mathbf{w}_1+\Delta\mathbf{w}t\end{aligned}$$

Z powodu notacyjnej, niech δw = d. Aby zachować ten sam widok, co zaproponowano wcześniej, można rozważyć ruch d jako inny segment. Następnie musisz obliczyć iloczyn krzyżowy (czyli wektor prostopadły) między d a normalnymi wektorami płaszczyzn {x, y, z} dla każdego pola. Dlatego musisz dodać dodatkowe sześć kierunkowych wektorów dla całkowitej listy 21:

$$\mathbf{l}=\left\{\begin{array}{l}\mathbf{x}_a,\mathbf{y}_a,\mathbf{z}_a, \\ \mathbf{x}_b,\mathbf{y}_b,\mathbf{z}_b, \\ \mathbf{x}_a\times\mathbf{x}_b,\mathbf{x}_a\times\mathbf{y}_b,\mathbf{x}_a\times\mathbf{z}_b, \\ \mathbf{y}_a\times\mathbf{x}_b,\mathbf{y}_a\times\mathbf{y}_b,\mathbf{y}_a\times\mathbf{z}_b, \\ \mathbf{z}_a\times\mathbf{x}_b,\mathbf{z}_a\times\mathbf{y}_b,\mathbf{z}_a\times\mathbf{z}_b, \\ \mathbf{d}\times\mathbf{x}_a,\mathbf{d}\times\mathbf{y}_a,\mathbf{d}\times\mathbf{z}_a, \\ \mathbf{d}\times\mathbf{x}_b,\mathbf{d}\times\mathbf{y}_b,\mathbf{d}\times\mathbf{z}_b\end{array}\right\}$$

Ta lista wektorów stanowi pełny test. Innymi słowy, jeśli wszystkie testy zakończą się niepowodzeniem, wykryto przecięcie i jeśli jeden test przejdzie, wykryto brak nonsekcji. Pozostaje tylko jeden problem. Nie znasz czasu przecięcia. To samo w sobie nie jest alarmujące, ponieważ wiesz, że jest pomiędzy ramkami. Możesz jednak znaleźć dokładny punkt kolizji między tymi dwoma obiektami. Oczywiście najwłaściwszą metodą osiągnięcia tego jest uzyskanie czasu. Możesz uzyskać czas, jeśli po prostu wyizolujesz równanie, którego używasz jako testu:

$$\begin{aligned}r_{ea}+r_{rb} &= \mathbf{l}\bullet\mathbf{w}(t) \\ &= \mathbf{l}\bullet(\mathbf{w}_1+\mathbf{d}t) \\ &= \mathbf{l}\bullet\mathbf{w}_1+\mathbf{l}\bullet\mathbf{d}t \\ t &= \frac{r_{ea}+r_{rb}-\mathbf{l}\bullet\mathbf{w}_1}{\mathbf{l}\bullet\mathbf{d}}\end{aligned}$$

Czas, w którym wydostaniesz się z tego równania, to czas, w którym projekcja dwóch obiekty z punktu przecięcia wektora kierunkowego l. Nie jest to domyślnie czas, w którym oba obiekty przecinają się same w sobie. Aby uzyskać rzeczywisty czas przecięcia, należy również sprawdzić następny wektor kierunkowy. Test przecięcia koniecznie musi wykryć kolizję, więc musisz ponownie obliczyć czas, ale musisz wziąć najlepszy czas między tymi dwoma. Następnie należy zastosować ten proces do każdej płaszczyzny i w konsekwencji obliczyć maksymalny czas przecięcia. Może wydawać się mylące, aby obliczyć maksymalny czas, aby obliczyć pierwszy moment, w którym zderzają się dwa obiekty, ale wynika to z samej definicji twierdzenia o osi rozdzielającej. Dopóki istnieje jedna istniejąca oś oddzielająca, obiekty nie przecinają się. Dlatego musisz kontynuować, dopóki nie zrobią tego wszystkie samoloty, a to może nastąpić tylko w maksymalnym czasie między zderzeniami. Teraz, gdy masz czas, pozostaje tylko obliczyć faktyczne skrzyżowanie. Jest to łatwe do zrobienia, jeśli śledzisz wektor kierunkowy skojarzony z ostatnim razem. Ten wektor kierunkowy jest wektorem, który uzyskuje najlepszy widok w domu na kolizję. Ponieważ jest to ostatni wektor, który powoduje zderzenie rzutowanych długości, jest to wektor, który widzi obiekty, gdy ich pierwsze wartości przecinają się. Ponownie, dla tej sekcji, przyjmuję podejście, które nieco różni się od poprzednich podejść. Zamiast próbować obliczyć położenie skrzyżowania, spróbuję ustalić, które wierzchołki skrzyni są tymi, które zderzają się z innym polem. Aby to zrobić, najpierw trzeba jeszcze raz zaobserwować związek między twierdzeniem o płaszczyźnie oddzielającej a twierdzeniem o osi rozdzielającej. Jeśli generujesz płaszczyznę, używając wektora kierunkowego jako wektora normalnego, możesz wygenerować ostatnią płaszczyznę, którą przecinasz. Kiedy już masz samolot, sztuczka polega po prostu na obliczeniu zestawu dla wierzchołków, które mają minimalną odległość między płaszczyznami. Zwróć uwagę, w jaki sposób tutaj użyto liczby mnogiej dla "wierzchołków". Musisz pamiętać, że dwa pola mogą przecinać się na wiele sposobów. Pojedynczy wierzchołek może się przecinać, ale dwie krawędzie również mogą się przecinać. Alternatywnie możesz również uzyskać przecięcie między dwiema płaszczyznami. Technika ta łatwo obejmuje każdy przypadek, ponieważ odległość między każdym z tych wierzchołków od płaszczyzny będzie względnie równa. Stosuje się tu względnie, ponieważ precyzja zmiennoprzecinkowa może sprawić, że odległości nie będą w praktyce dokładnie równe. W związku z tym trzeba porównać przy użyciu progu. Ponieważ obliczamy odległość między płaszczyzną a wierzchołkami w czasie t, nie jest tak naprawdę ważne, gdzie płaszczyzna się znajduje, pod warunkiem, że płaszczyzna przecina się tylko z obiektem na powierzchni. Innymi słowy, wszystkie wierzchołki muszą znajdować się po obu stronach płaszczyzny, inaczej odległość między płaszczyzną a wierzchołkami nie ma znaczenia. Wiesz z hipotezy, że początkowo pudełka się nie przecinają. W związku z tym wystarczy wybrać punkt z drugiego pola jako punkt zakotwiczenia, a będziesz mieć pewność, że wszystkie wierzchołki z drugiego pola będą po jednej stronie płaszczyzny. Tak więc, płaszczyzna, o której mowa, przez ostatni wektor kierunkowy l i dowolny punkt p z pola B, gdy testuje się względem pola A, jest twierdzeniem o osi rozdzielającej, ponieważ umożliwia pracę z różnymi obiektami . Może nie być tak bezpośredni jak niektóre inne metody przedstawione w tym rozdziale, ale oferuje solidną podstawę, która działa dla dowolnego wypukłego ciała (ciało, które nie ma żadnych wgnieceń). Specyficzny przykład kolizji pudełkowej został rozwiązany, ponieważ gdy zrozumiesz ideę wykrywania kolizji w pudełku przy użyciu twierdzenia o osi rozdzielającej, nie będziesz miał problemu ze zrozumieniem trójkąta trójkąta, trójkąta i wielu innych podobne rodzaje testów przecięć. Jedyną różnicą, na przykład, jeśli pracujesz z trójkątem, jest to, że zakres w samolocie będzie nieco inny, i masz tylko jedną płaszczyznę do martwienia się zamiast trzech.

Wybieranie

Kiedy prowadzisz grę taką jak Warcraft III, musisz mieć możliwość wybrania niektórych jednostek poruszających się po ekranie za pomocą kliknięcia myszą. Oczywiście nie można po prostu machać czarodziejską różdżką i oczekiwać, że gra będzie wiedziała, który obiekt wybrać. Na szczęście, wybieranie w grach 3D jest stosunkowo prostym zadaniem i nie musi zajmować się pojęciem czasu. W związku z tym nie musi zbytnio zagłębiać się w złożone algorytmy. Wszystkie problemy z kompletacją można zredukować do nieskończonego przecięcia linii z obiektem na ekranie. Aby rozwiązać problem z wybieraniem, gdy użytkownik kliknie myszą w określonej lokalizacji na ekranie, musisz przekształcić tę współrzędną <x, y> we współrzędne ekranu na wektor 3D, który wskazuje od kamery w głąb świata. Pierwszym krokiem jest zrozumienie, jak odzyskać taki wektor.

Obliczanie pozycji i kierunku pobrania

Pobieranie kierunku i położenia kilofa jest dość trywialne. Pozycja picka (lub zakotwiczonego wierzchołka) jest po prostu położeniem kamery. Wektor przechodzi przez środek kamery, więc jest to punkt w twoim zbiorze wektorowym. Trochę trudniej jest odzyskać kierunek kilofa. Proces jest dość prosty, ponieważ wystarczy spojrzeć na to, jak współrzędna 3D może stać się współrzędną 2D, aby zrozumieć, jak powinna działać odwrotność tego procesu. Pracując w tył, ostatnią rzeczą, jaką robią interfejsy API 3D, jest zamiana transformowanych współrzędnych na współrzędne ekranu. W związku z tym pierwszą czynnością, jaką powinieneś wykonać dla swojego wektora 2D, jest skalowanie go do zakresu <x, y> z zakresu [-1, 1]. Ten proces może już być wykonany, w zależności od używanego interfejsu API. Kiedy już to zrobisz, musisz cofnąć pracę wykonaną przez macierz projekcji. W związku z tym należy obliczyć odwrotność macierzy projekcji. Jeśli weźmiesz macierz perspektywiczną pokazaną w części 5, "Transformacje", przy zachowaniu dokładnych wartości terminów abstrakty, otrzymasz następującą macierz odwrotną:

$$\begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ c & d & e & -1 \\ 0 & 0 & f & 0 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{a} & 0 & 0 & 0 \\ 0 & \frac{1}{b} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{f} \\ \frac{c}{a} & \frac{d}{b} & -1 & \frac{e}{f} \end{bmatrix}$$

Zatem nie jest konieczne przechodzenie przez typową matrycę odwrotną, po której następuje mnożenie wektorowe. Zamiast tego możesz po prostu podzielić x przez a i podzielić y przez b, a następnie przydzielić z = -1. Pierwszym krokiem (z pominięciem transformacji obiektu) jest matryca transformacji kamery i świata, więc twoim ostatnim krokiem jest odwrócenie procesu. Jeśli pozwolisz macierzy na M, wszystko, co musisz zrobić w tym momencie, to odwrócić matrycę i obliczyć iloczyny macierzy z wektorem, który do tej pory zbudowałeś. Aby to działało, musisz mieć pewność, że matryca światowa została załadowana do stosu API i pobrać tę macierz. Macierz świata nie powinna być sprzężona z macierzą obiektów, ponieważ zbieranie powinno odbywać się w skali światowej, a nie w perspektywie obiektu. Last but not least, musisz odjąć pozycję kamery od wektora tak, aby uzyskać tylko kierunek. Pozycja kamery powinna być przechowywana w oddzielnym wektorze, abyś mógł łatwo wyrazić równanie nieskończonej linii $x(t) = x_0 + vt$. Jeśli rozumiesz stos macierzy 3D API, nie powinno to stanowić problemu. Wszystkie działają tak samo. Spójrz na kod źródłowy, aby rzucić okiem na specyfikę OpenGL. W tej sekcji omówiono niektóre dość trywialne przypadki w porównaniu do tego, co zobaczyłeś. Jeśli szukasz testu na przecięcie promienia trójkąta lub przecięcia promienia, powinieneś przyjrzeć się Części 4, która zawiera wszystkie szczegółowe szczegóły o takiej realizacji.

Przecięcie Promień-Kula

Bez wątpienia jest to jeden z najłatwiejszych testów do wykonania. Z zakresu dotychczasowych technik, ten typ problemu powinien być bezużyteczny. Wszystko, co musisz zrobić, to obliczyć minimalną odległość między punktem p, reprezentującym środek kuli i sprawdzić, czy promień jest mniejszy lub równy temu. Oczywiście, chcesz być nieco bardziej optymalny niż test pokazany w rozdziale 4, abyś mógł obliczyć kwadrat każdej strony i sprawdzić, czy nierówność utrzymuje się. Matematycznie, ten trywialny test jest przykładany do środka sfery v o promieniu ri do promienia równania p + tl w następujący sposób:

$$r^2 \le (v_x - p_x)^2 + (v_y - p_y)^2 + (v_z - p_z)^2 - \frac{((\mathbf{v}-\mathbf{p})\bullet \mathbf{l})^2}{\|\mathbf{l}\|_2^2}$$

Przecięcie Promień – Pole

Pole, w przypadku płaszczyzny, było traktowane podobnie jak przypadek płaszczyzny i sfery. Ten test nie jest wyjątkiem. Jedynym trudnym aspektem tego testu jest to, że nie masz od razu wektora, na którym można wyświetlać pole w celu obliczenia efektywnych promieni. Ale oczywiście możesz obliczyć ten wektor. Dopóki pamiętasz, że wektor musi być normalny i że ten normalny wektor przechodzi od punktu na linii do wierzchołka v, wszystko staje się proste:

$$\begin{aligned} 0 &= \mathbf{l} \bullet \mathbf{n} \\ &= \mathbf{l} \bullet (\mathbf{p} + t\mathbf{l} - \mathbf{v}) \\ &= \mathbf{l} \bullet (\mathbf{p} - \mathbf{v}) + t\mathbf{l} \bullet \mathbf{l} \\ t &= \frac{\mathbf{l} \bullet (\mathbf{p} - \mathbf{v})}{\mathbf{l} \bullet \mathbf{l}} \end{aligned}$$

Gdy masz czas, skończysz. Możesz go zastąpić równaniem, aby znaleźć wartość n, a następnie obliczyć efektywny promień pola. Dzięki efektywnemu promieniu możesz zastosować ten sam test, aby zweryfikować skrzyżowanie. Poniższe równanie pokazuje to:

$$\mathbf{n} = \mathbf{p} - \mathbf{v} + \frac{\mathbf{l} \bullet (\mathbf{p} - \mathbf{v})}{\mathbf{l} \bullet \mathbf{l}} \mathbf{l}$$

$$r_e = |\mathbf{x} \bullet \mathbf{n}| + |\mathbf{y} \bullet \mathbf{n}| + |\mathbf{z} \bullet \mathbf{n}|$$

$$r_e^2 \le (v_x - p_x)^2 + (v_y - p_y)^2 + (v_z - p_z)^2 - \frac{((\mathbf{v}-\mathbf{p})\bullet \mathbf{l})^2}{\|\mathbf{l}\|_2^2}$$

Przecięcie Promień - Kapsula

Oto nowy interesujący kształt, którego być może wcześniej nie widziałeś. Kapsułka jest bardzo podobna do cylindra i wygląda trochę jak pigułka. Matematycznie kapsuła jest przedmiotem, dla którego każdy punkt na jego powierzchni jest w równej odległości od linii. W konsekwencji na osiach prostopadłych do linii obiekt wygląda jak walec, ale końce są półkulami, jak pokazano na rysunku 10.16.

Ze względu na dość ładną definicję matematyczną, jest to obiekt, który nie jest zbyt trudny do sprawdzenia pod kątem kolizji. Zacznijmy od zdefiniowania kapsuły jako dwóch wierzchołków tworzących jej linię, u, v i promień r. Tak więc linia środkowa kapsuły ma równanie z zakresem t = [0, 1]:

f(t) = u + t(v – u)

Pierwszym łatwym do wykonania testem jest obliczenie kwadratu odległości między linią a promieniem. Nauczyłeś się już, jak to się robi w sekcji dotyczącej wykrywania kolizji kuli, więc nie ma sensu ponownie przez nią przechodzić. Najpierw obliczasz czas przecięcia dla dwóch nieskończonych linii:

$$\mathbf{a} = \mathbf{u} - \mathbf{p}$$
$$\mathbf{b} = \mathbf{v} - \mathbf{u} - \mathbf{w}$$
$$t = -\frac{\mathbf{a} \bullet \mathbf{b}}{\mathbf{b} \bullet \mathbf{b}}$$

Następnie należy przyciąć czas t tak, aby nie przekroczył jego zakresu [0, 1]. Kiedy już to zrobisz, musisz zastąpić czas z powrotem w równaniu na odległość, jak pokazano w przypadku kolizji sfery kuli, aby uzyskać końcowy test:

$$r^2 \leq \mathbf{a} \bullet \mathbf{a} + 2\mathbf{a} \bullet \mathbf{b}t + \mathbf{b} \bullet \mathbf{b}t^2$$

Ta część, podobnie jak wiele innych, może zostać rozszerzona do całej książki. Przypadki, które masz przestudiowane tutaj są konkretne, przydatne, które można spotkać z perspektywy gry. Tak jak zawsze bardzo dobrze jest poznać proces myślenia za nimi, abyś mógł obliczyć własne testy. Na przykład, powiedzmy, że musisz wykonać test, który wykrywa kolizję między płaszczyzną a elipsoidą. Elipsoida jest sferą, czym jest elipsa do koła. Tak więc elipsoida jest opisana przez trzy promienie i jest to naprawdę tylko elipsa skalowana inaczej w x, yi z. Jeśli miałbyś rozwiązać ten problem w inteligentny sposób, po prostu zastosowałbyś skalowalną macierz transformacji do płaszczyzny sfery tak, aby elipsoida stała się kulą. Przekształciłoby to problem, który wydawał się dość skomplikowany w inny problem, dla którego już masz rozwiązanie. Jeśli przekonwertujesz problem na taki, jaki znasz, masz już rozwiązanie, o ile odwrócisz zmianę, którą zrobiłeś na końcu, aby nie zmieniać głównego problemu. W takim przypadku, po obliczeniu pozycji kolizji, należy zastosować odwrotną macierz transformacji, aby przekształcić ze skalowanych współrzędnych na współrzędne globalne.