

Zajęcie 3. ĆWICZENIA W MODELOWANIU MATERIAŁÓW O RÓŻNYCH WŁAŚCIWOŚCIACH FIZYCZNYCH

CELE:

1. Opracowanie gry komputerowej na podstawie komponentów Physics
2. Obiekt gry Terrain
2. Importowanie do projektu paczek sklepu Asset Store – Tangy Texture

Link: http://wiki.unity3d.pl/doku.php?id=unity_manual:physics

Fizyka

Unity posiada wbudowany silnik fizyki **NVIDIA PhysX**. Pozwala to na zachodzenie unikalnych zachowań i posiada wiele przydatnych funkcji.

Podstawy

Aby umieścić obiekt pod kontrolą fizyki, po prostu wystarczy dodać **Rigidbody** do niego. Gdy to zrobisz, na obiekt będzie miała wpływ grawitacja i mogą się zderzyć z innymi obiektami na świecie.

Rigidbody

Rigidbody fizycznie zymuluje obiekty. Możesz użyć Rigidbody do rzeczy, które gracz może popychać wokoło, na przykład skrzynie lub luźne przedmioty lub może poruszać Rigidbody bezpośrednio przez dodanie sił do niego przez wykonywanie skryptów.

Jeśli przeniesiesz bezpośrednio Transform z non-Kinematic Rigidbody to może nie zderzać się poprawnie z innymi obiektami. Zamiast tego należy przenieść Rigidbody przy zastosowaniu siły i momentu obrotowego do niego. Można również dodać **Joints** do Rigidbody aby zachowanie było bardziej złożone. Na przykład można zrobić drzwi fizyczne lub żuraw z kołyszącym się łańcuchem.

Można również użyć Rigidbody by pobudzić pojazdy do życia, na przykład możesz stworzyć samochód używając Rigidbody, 4 **Wheel Colliders** i skrypt stosowania siły koła w oparciu o **Input** użytkownika.

Samoloty można zrobić przy zastosowaniu siły do Rigidbody ze skryptu. Lub można utworzyć specjalne pojazdy lub roboty przez dodanie różnych Joints i stosowania siły za pośrednictwem skryptów.

Rigidbody jest najczęściej używany w połączeniu z **prymitywnymi collider-ami**

Porady:

- Rodzic i dziecko nie powinni mieć ze sobą wspólnego rigidbody
- Nigdy nie należy skalować nadrzędnego rigidbody

Kinematic Rigidbody

Kinematic Rigidbody jest to Rigidbody, który ma włączoną opcję **Is Kinematic**. Kinematic Rigidbody nie dotyczy siły, grawitacji lub kolizji. Są wyraźnie prowadzone przez ustawienie pozycji i rotacji przekształcenia lub animacji, jeszcze ich interakcji z innymi nie - Kinematic Rigidbody.

Kinematic Rigidbody poprawnie **pobudzi** inny Rigidbody, kiedy zderzają się z nimi, a one dotyczą tarcia Rigidbody umieszczonych na nich.

Oto kilka przykładowych zastosowań Kinematic Rigidbody:

1. Czasami chcesz obiekt, który ma być pod kontrolą fizyki, ale w innej sytuacji wyraźnie sterowane ze skryptu lub animacji. Na przykład, można zrobić animowaną postać, której joints Rigidbody dołączone i związane z stawami do wykorzystania jako Ragdoll. Większość czasu postać jest pod kontrolą animacji, więc możesz zrobić Kinematic Rigidbody. Ale gdy dostaje uderzenie chcesz go przekształcić w Ragdoll i odczuć fizykę. Aby to osiągnąć, po prostu wystarczy wyłączyć właściwość **Is Kinematic**.

Statyczne Collider-y

Static Collider jest to GameObiekt który posiada **Collider** ale nie Rigidbody. Static Collider są wykorzystywane do poziomu geometrii, który zawsze pozostaje w tym samym miejscu i nigdy nie porusza się. Możesz dodać **Mesh Collider** do swojego już istniejącego mesh-a (nawet lepsze wykorzystanie pole wyboru **Import Settings** generowania Collider-ów) lub można użyć jednego z innych typów Collider-ów.

Nigdy nie należy przenosić Static Collider na zasadzie klatka po klatce. Przeniesienie Static Collider spowoduje wewnętrzne przeliczenie w PhysX to dość ciężkie i co spowoduje duży spadek wydajności. Na początku, że zachowanie budzi inne Rigidbody w oparciu statycznych Collider-ów jest niezdefiniowany i przeniesienie Static Collider nie stosuje się tarcia do Rigidbody, żeby go dotykać.

Kontroler postaci

Używasz **Character Controller** jeśli chcesz tworzyć humanoidalne postacie. Może to być główny bohater z trzeciej osoby, strzelanka FPS lub postać wroga.

Te kontrolery nie przestrzegają zasad fizyki, ponieważ nie czują się w tym dobrze. Zamiast tego Character Controller wykonuje wykrywanie kolizji aby upewnić się, że postać można przesuwając wzdłuż ściany, w górę i w dół po schodach, itp.

Character Controller nie dotyczą siły, ale można je wcisnąć Rigidbody przy zastosowaniu siły do nich ze skryptu. Zwykle wszystkie postacie humanoidalne są implementowane przy użyciu Character Controllers.

Character Controller są z natury unphysical, więc jeśli chcesz zastosować fizykę - huśtawka na linie, dostaniesz wciśnięty przez duże skały - do swojej postaci, musisz użyć Rigidbody, to pozwoli Ci używać stawów i sił na swoją postać. Character Controller są zawsze dostosowane wzdłuż osi Y, więc należy użyć Rigidbody, jeśli Twoja postać musi być w stanie zmieniać orientację w przestrzeni (np. w ramach zmiany grawitacji). Należy jednak pamiętać, że dostrajanie Rigidbody do postaci jest trudne ze względu na sposób unphysical, w którym powinna zachowywać się postać w grze. Inną różnicą jest to, że Character Controller można przesunąć gładko przez kroki o określonej wysokości, a Rigidbody nie.

Jeśli rodzic Character Controller z Rigidbody otrzymasz zachowanie „Joint”

Rigidbody

Rigidbody włącza GameObiekty GameObjects do działania pod kontrolą fizyki. Z Rigidbody można otrzymać siły i momenty obrotowe do stworzenia obiektu poruszającego się w realistyczny sposób. Wszelkie GameObiekt musi zawierać Rigidbody aby być pod wpływem grawitacji, działa na podstawie dodania sił za pośrednictwem skryptów lub interakcję z innymi za pośrednictwem silnika fizyki **NVIDIA PhysX**.

Rigidbodyes pozwala GameObiektowi do działania pod wpływem fizyki

Właściwości

- **Mass** - Masa obiektu (dowolnych jednostek). Zaleca się, aby mas nie była większa mniej więcej niż 100 razy niż w innych Rigidbody.
- **Drag** - Wartość oporu powietrza która wpływa na obiekt, przemieszczając się z siłą. 0 oznacza brak oporu powietrza sprawia, że obiekt zatrzymuje się natychmiast.
- **Angular Drag** - Wartość opór powietrza która wpływa na obiekt, podczas obracania od momentu obrotowego. 0 oznacza brak oporu powietrza i sprawia, że obiekt zatrzymuje się o obrocie natychmiast.
- **Use Gravity** - Jeśli włączone, obiekt ma wpływ grawitacji.
- **Is Kinematic** - Jeśli włączone, obiekt nie będzie napędzany przez silnik fizyki, a tylko mogą być manipulowane przez jego przekształcenia. Jest to przydatne na ruchomych platformach lub jeśli chcesz animować Rigidbody, który ma dołączony **Hinge Joint**.
- **Interpolate** - Spróbuj jednej z opcji tylko wtedy, gdy widzisz szarpanie w ruch swojego Rigidbody.
- **None** - Zastosowano nie interpolacji.
- **Interpolate** - Przekształcenie jest wygładzone, oparte na przekształceniu poprzedniej ramki.
- **Extrapolate** - Przekształcenie jest wygładzone, w oparciu o szacunkową transformację następnej klatki.
- **Collision Detection** - Stosowany w celu zapobiegania szybko poruszającym się obiektom z przechodzeniem przez inne obiekty bez wykrywania kolizji.
- **Discrete** - W scenie, należy użyć dyskretnego wykrywania kolizji w stosunku do wszystkich innych collider-ów. Inne collider-y używają dyskretnego wykrywania kolizji, podczas badania zderzenia przeciwko niemu. Używane do normalnej kolizji (jest to wartość domyślna).
- **Continuous** - Za pomocą dyskretnego wykrywania kolizji przeciwko dynamicznemu collider-owi (z rigidbody) i wykrywanie kolizji ciągłe przeciwko statycznemu MeshCollider (bez rigidbody). Rigidbody ustawiony na Continuous Dynamic użyje wykrywania kolizji ciągłego (Continuous), podczas badania kolizji przed tym rigidbody. Inne rigidbody użyje wykrywania kolizji dyskretnego (Discrete). Używane dla obiektów, które wymaga wykrywania Continuous Dynamic i zderzają się ze sobą. (To ma duży wpływ na wydajność fizyki, to ustaw Discrete, jeśli nie masz problemów z kolizjami szybkich obiektów)
- **Continuous Dynamic** - Za pomocą wykrywania kolizji ciągłego i ustawieniu ciągłego i ciągłego dynamicznego kolizji obiektów. To będzie również używać wykrywania kolizji ciągłego przeciwko

statycznemu MeshColliders (bez rigidbody). Dla wszystkich innych collider-ów używa wykrywanie kolizji dyskretnego. Używane dla szybko poruszających się obiektów.

- **Constraints** - Ograniczenia w ruchu Rigidbody
- **Freeze Position** - Zatrzymuje ruch Rigidbody w świecie selektywnie na osi X, Y i Z.
- **Freeze Rotation** - Zatrzymuje obracanie Rigidbody w świecie selektywnie na osi X, Y i Z.

Szczegóły

Rigidbody umożliwia GameObiektom działania pod kontrolą silnika fizyki. To otwiera bramę do realistycznych kolizji. Manipulując swoimi GameObiektami dodając siły Rigidbody tworzy zupełnie inny styl i wygląd niż dostosowanie i przekształcenie części bezpośrednio. Ogólnie rzecz biorąc nie należy manipulować Rigidbody i Transform samego GameObiektu - tylko jedno lub drugie.

Największą różnicą pomiędzy manipulowaniem Transform w funkcji Rigidbody jest wykorzystanie sił. Rigidbody może odbierać siły i moment obrotowy, ale Transform nie może. Transform można przesuwać i obracać, ale nie jest to samo co fizyka. Zauważysz wyraźną różnicę, kiedy to wypróbujesz. Dodawanie siły/momentu obrotowego do Rigidbody będzie faktycznie zmienić położenie obiektu i obrót elementu Transform. Dlatego należy używać tylko jednego lub drugiego. Zmiana Transform podczas korzystania z fizyki może być przyczyną problemów z kolizjami i innych obliczeń.

Rigidbody musi być jawnie dodane do twójego GameObiektu zanim zostaną pod wpływem silnika fizyki. Możesz dodać Rigidbody do wybranego obiektu za pomocą paska menu **Components**→**Physics**→**Rigidbody**. Teraz twój obiekt będzie gotowy do fizyki. To spowoduje spadanie pod wpływem grawitacji i otrzyma siły za pośrednictwem skryptów, ale może dodać **Collider**, aby zachowywał się dokładnie, jak chcesz.

Parenting

Gdy obiekt jest pod kontrolą fizyki, porusza się w sposób semi-independently i przenosi przekształcenia rodzica. Jeśli przeniesiesz jakiegokolwiek rodzica, będą one ciągnąć za sobą Rigidbody dziecka razem za nim. Jednak Rigidbody będzie nadal spadać z powodu grawitacji oraz reagowania na zderzenia kolizji.

Scripting

Kontrolować swoje Rigidbody, będziesz przede wszystkim wykorzystując i dodając skrypty siły lub momentu obrotowego. Można to zrobić poprzez wywołanie **AddForce()** i **AddTorque()** w obiekcie z Rigidbody. Pamiętaj, że nie powinno być bezpośrednio zmieniane Transform obiektu podczas korzystania z fizyki.

Animacja

W niektórych sytuacjach, głównie tworzenia efektów ragdoll, potrzebne jest przełączanie kontroli obiektu pomiędzy animacjami i fizyką. W tym celu Rigidbody mogą być oznaczone jako **Is Kinematic**. Podczas gdy Rigidbody jest oznaczony jako **Is Kinematic**, nie będzie zachodzić kolizja, siły lub jakiegokolwiek innej części PhysX. Oznacza to, że trzeba będzie sterować obiektem poprzez manipulowanie składnikiem **Transform**. Kinematic Rigidbody wpłynie na inne obiekty, ale sam nie będzie pod wpływem fizyki. Na przykład Joints, które są dołączone do obiektów Kinematic będzie ograniczać inne Rigidbody dołączone do nich i Kinematic Rigidbody wpłynie na inne Rigidbody za pomocą kolizji.

Collider-y

Collider-y są innego rodzaju składnika, które muszą zostać dodane wraz z Rigidbody w celu umożliwienia wystąpienia kolizji. Jeśli dwa Rigidbody wpadną na siebie, silnik fizyki nie obliczy

kolizji, chyba, że oba obiekty mają również dołączony Collider. Rigidbody będą po prostu przechodzić przez siebie podczas symulacji fizyki jeśli nie będzie dodany Collider.



Collider określa fizyczne granice Rigidbody

Dodawanie Collider-ów za pomocą menu **Component-Physics**. Zobacz na odnośnikach składniki wszelkich indywidualnych Collider-ów, bardziej szczegółowe informacje:

- **Box Collider** - pierwotny kształt sześcianu
- **Sphere Collider** - pierwotny kształt kuli
- **Capsule Collider** - pierwotny kształt kapsułki
- **Mesh Collider** - tworzy Collider z siatki obiektu, nie koliduje z innym Mesh Collider
- **Wheel Collider** - specjalnie dla tworzenia samochodów lub innych pojazdów w ruchu

Złożone Collider-y

Compound Colliders (Złożone Collider-y) są kombinacją prymitywnych koliderów, wspólnie działających jako pojedynczy kolider. Przydaje się gdy masz kompleks siatek do użycia Collider-ów, ale nie możesz użyć **Mesh Collider**. Aby utworzyć złożony kolider należy utworzyć obiekt dla twojej kolizji obiektu, następnie dodać prymitywne Collider-y do każdego obiektu podrzędnego.

Rzeczywisty złożony Collider

Na zdjęciu powyżej GameObject Gun Model ma Rigidbody, w załączeniu i wiele prymitywnych Collider-ów jako GameObiekty dzieci. Gdy rodzic Rigidbody jest przesuwane siłą, dziecko Collider przeniesie się razem z nim. Prymitywne Collider-y zderzają się z Mesh Collider-ami i rodzic Rigidbody zmienia sposób, w jaki porusza się w oparciu o siły stosowane i jak jej dziecko Collider-y wchodzi w interakcję z innymi Collider-ami na scenie.

Mesh Collider nie mogą normalnie zderzać się ze sobą. Jeśli Mesh Collider jest oznaczony jako **Convex**, wtedy to koliduje z innym Mesh Collider-ami. Dobrym rozwiązaniem jest użycie prymitywnych Collider-ów na wszelkie obiekty, które poruszają się i Mesh Collider dla obiektów statycznych, tła.

Detekcja Continuous Collision

Wykrywanie Continuous Collision jest funkcją, zapobiega szybkim kolizjom, przechodzących się nawzajem. To może się zdarzyć, gdy używasz normalnego wykrywania kolizji (**Discrete**), gdy obiekt jest po jednej stronie Collider-a w jednej klatce, a już minął Collider w następnej klatce. Aby rozwiązać ten problem, można włączyć wykrywanie kolizji ciągłe na rigidbody szybkich obiektów. Ustaw tryb wykrywania kolizji na **Continuous**, aby zapobiec rigidbody przechodzenie przez statyczne (tzn. nie rigidbody) Mesh Collider-y. Ustaw go na **Continuous Dynamic** również uniemożliwia rigidbody przechodzić przez wszelkie inne obsługiwane rigidbody z wykrywanie kolizji tryb **Continuous** lub **Continuous Dynamic**. Wykrywanie ciągłe kolizji jest obsługiwana dla Box, Sphere i Capsule Collider. Należy zauważyć, że wykrywanie kolizji ciągłe służy jako zabezpieczenie złapania kolizji w przypadkach, gdy obiekty inaczej by przeszły przez siebie, ale nie dostarczyły kolizji fizycznie dokładnego wyniku, więc można jeszcze rozważyć zmniejszenie ustalonej wartości kroku czasu w TimeManager inspektora do symulacji bardziej dokładnej, jeśli napotkasz na problemy z szybko poruszającymi się obiektami.

Używanie odpowiedniego rozmiaru

Rozmiar siatki twojego GameObiektu jest znacznie ważniejszy niż masa Rigidbody. Jeśli uważasz, że twój Rigidbody nie działa dokładnie, jak można oczekiwać - porusza się powoli, pływa, lub nie zderza się poprawnie - rozważyć dostosowanie skali siatki mesh-u swojego assetu. Unity w domyślnej skali jednostki jest 1 unit = 1 metr, więc skala importowanych mesh-y jest utrzymywana i stosowana do obliczeń fizyki. Na przykład rozpadający się wieżowiec będzie rozpadał się zupełnie inaczej niż wieża z zabawkowych bloków, więc powinny być wzorowane obiekty o dokładnie różnej skali.

Jeśli są modelowani ludzie, upewnij się, że są wysocy na około 2 metrów wysokości w Unity. Można sprawdzić czy obiekt ma odpowiedni rozmiar w porównaniu do domyślnego **Cube**. Można utworzyć Cube przy użyciu **GameObject → Create Other → Cube**. Wysokość Cube będzie dokładnie 1 metr, więc twój człowiek powinno być dwa razy tej wysokości.

Jeśli nie jesteś w stanie dostosować skali siatki, możesz zmienić jednolitą skalę danej siatki assetu wybierając go w **Widoku Projektu** i zmieniając **Assets → Import Settings...** z paska menu. Tutaj, możesz zmienić skalę i ponownie zaimportować siatki.

Jeśli twija gra wymaga aby twój GameObiekt musi być utworzony w różnych skalach, to dobrze dopasuj wartość twojej skali Transform osi. Minusem jest to, że symulacja fizyki musi zrobić więcej pracy w tym czasie obiekt zostanie uruchomiony i może spowodować spadek wydajności w grze. To nie są straszne straty, ale nie jest tak skuteczne jak kończy swoją skalę w pozostałych dwóch opcjach. Należy pamiętać, że niejednolite skale mogą spowodować niepożądane zachowania/efekty, gdy rodzicielstwo jest wykorzystywane również rodzicielstwo. Z tych powodów jest zawsze najlepiej utworzyć obiekt w poprawnej skali w modelowaniu.

Wskazówki

- Względna **Mass** dwóch Rigidbodies określa, jak reagują, kiedy zderzają się ze sobą.
- Co jeden Rigidbody ma większe **Mass** niż inny nie sprawiają że szybciej spada w swobodnym spadku. Korzysta za to z **Drag**.
- Niska wartość **Drag** sprawia, że obiekty wydają się ciężkie. Wysoki jeden sprawia, że wydaje się lekki. Typowe wartości oporu są między *.001 (bryły metalu) i 10 (pióro).
- Jeśli manipulujesz bezpośrednio składnikiem Transform obiektu ale nadal chcesz fizyki dołączyć Rigidbody i ustaw tryb kinematyczny.
- Jeśli przenosisz GameObiekt poprzez jego składnik Transform, ale chcesz otrzymywać wiadomości kolizji/wyzwalania, możesz dołączyć Rigidbody do obiektu, który jest w ruchu.

Stała siła

Constant Force jest szybkie narzędzie do dodawania stałej siły do Rigidbody. To działa świetnie na jeden strzał obiektów, takich jak rakiety, Jeśli nie chcesz, aby rozpocząć z dużą prędkością lecz przyspieszał.

Rakiety napędzana do przodu Constant Force

Właściwości

- **Force** - wektor siły do stosowany w przestrzeni świata.
- **Relative Force** - wektor siły stosowany w lokalnej przestrzeni obiektu.

- **Torque** - wektor momentu obrotowego stosowany w przestrzeni świata. Obiekt rozpocznie obracać się wokół tego wektora. Dłuższy wektor większa szybkość obrotu.
- **Relative Torque** - wektor momentu obrotowego, stosowany w lokalnej przestrzeni. Obiekt rozpocznie obracać się wokół tego wektora. Dłuższy wektor większa szybkość obrotu.

Szczegóły

Rakieta, która przyspiesza do przodu należy ustawić **Relative Force** wzdłuż osi dodatniej. Następnie użyj w Rigidbody właściwości **Drag**, aby nie przekraczała maksymalnej prędkości (im wyższa wartość tym niższa prędkość maksymalna). W Rigidbody także upewnij się czy wyłączyłeś grawitację, tak że rakieta będzie zawsze zatrzymać się na swojej drodze.

Wskazówki

- Aby obiekt płynął do góry dodaj Constant Force z właściwością **Force** o wartości dodatniej Y.
- Aby obiekt leciał do przodu, dodaj Constant Force z właściwością **Relative Force** o wartości dodatniej Z.

Sphere Collider

Sphere Collider jest podstawowym prymitywnym koliderem w kształcie kuli.

Stos Sphere Colliders

Właściwości

- **Is Trigger** - jeśli włączone, Collider jest używany do wyzwalania zdarzenia i jest ignorowany przez silnik fizyki.
- **Material** - Odniesienie do **Fizycznego Materiału**, który określa, jak ten Collider współdziała z innymi.
- **Radius** - Rozmiar Collider-a.
- **Center** - Pozycja Collider-a w lokalnej przestrzeni obiektu.

Szczegóły

Sphere Collider może być przeskalowywany w jednolitej skali nie wzdłuż poszczególnej osi. To świetna sprawa dla spadających głązów, piłki ping pong, kulki, itp.

Standardowy Sphere Collider

Collider współpracuje z Rigidbody wnosząc do fizyki w Unity życie. Mając na uwadze Rigidbody pozwala aby obiekty były pod kontrolą fizyki, Collider-y natomiast aby obiekty mogły zderzać się ze sobą. Collider mogą być dodawane do obiektu niezależnie od Rigidbody. Collider-y nie konieczne potrzebują Rigidbody w załączeniu, ale Rigidbody musi być dołączone w celu, aby obiekt przeniósł się w wyniku kolizji.

W przypadku wystąpienia kolizji między dwoma Collider-ami i jeśli przynajmniej jeden z nich ma dołączony Rigidbody, **trzy wiadomości kolizji** są wysyłane do obiektów dołączonych do nich. Te zdarzenia mogą być obsługiwane w skryptach i pozwalają tworzyć unikalne zachowania z lub bez korzystania z wbudowanego silnika NVIDIA PhysX.

Triggers (wyzwalacze)

Alternatywny sposób przy użyciu Collider-ów jest aby oznaczyć je jako **Trigger**, wystarczy zaznaczyć pole właściwość **Is Trigger** w Inspektorze. Trigger-y skutecznie są ignorowane przez silnik fizyki i ma unikalny zestaw **trzech wiadomości kolizji** które są wysyłane, gdy występuje kolizja z Trigger-ami. Wyzwalacze są przydatne do uruchomienia innych wydarzeń w grze, jak przerywniki, otwierające się automatycznie drzwi, wyświetlanie wiadomości poradnika, itp. Użyj wyobraźni!

Należy pamiętać, aby doszło do zderzenia, gdy zderzają się dwa wyzwalacze, jeden z nich musi zawierać również Rigidbody. Trigger zderzają się z normalnie z Collider-ami jeden z nich musi mieć dołączone Rigidbody. Zobacz szczegółowy schemat różnego rodzaju kolizji, działań macieży kolizji w sekcji **Zaawansowane** poniżej.

Tarcie i odbijanie

Tarcie, odbijanie i miękkość są zdefiniowane w **Fizycznych Materiałach, Standardowe Assety** zawierają najczęściej fizyczne materiały. Aby użyć jednego z nich kliknij na menu rozwijalne Physics Material i wybierz jeden, np. Ice. Można, także **tworzyć** własne materiały fizyczne i dostosować wszystkie wartości tarcia.

Złożone Collider-y

Compound Colliders (Złożone Collider-y) są kombinacją prymitywnych koliderów, wspólnie działających jako pojedynczy kolider. Przydaje się gdy masz kompleks siatek do użycia Collider-ów, ale nie możesz użyć **Mesh Collider**. Aby utworzyć złożony kolider należy utworzyć obiekt dla twojej kolizji obiektu, następnie dodać prymitywne Collider-y do każdego obiektu podrzędnego.

Rzeczywisty złożony Collider

Na zdjęciu powyżej GameObject Gun Model ma Rigidbody, w załączeniu i wiele prymitywnych Collider-ów jako GameObiekty dzieci. Gdy rodzic Rigidbody jest przesuwane siłą, dziecko Collider przeniesie się razem z nim. Prymitywne Collider-y zderzają się z Mesh Collider-ami i rodzic Rigidbody zmieni sposób, w jaki porusza się w oparciu o siły stosowane i jak jej dziecko Collider-y wchodzi w interakcję z innymi Collider-ami na scenie.

Mesh Collider nie mogą normalnie zderzać się ze sobą. Jeśli Mesh Collider jest oznaczony jako **Convex**, wtedy to koliduje z innym Mesh Collider-ami. Dobrym rozwiązaniem jest użycie prymitywnych Collider-ów na wszelkie obiekty, które poruszają się i Mesh Collider dla obiektów statycznych, tła.

Wskazówki

- Aby dodać wiele Collider-ów dla obiektu, utwórz dziedziczone GameObiekt i dołącz Collider-y do każdego z nich. Dzięki temu każdym Collider-em manipulujesz niezależnie.
- Można spojrzeć na gizma w Widoku Sceny, aby zobaczyć, jak Collider-ów jest na obiekcie.
- Dołóż wszelkich starań aby dopasować Collider-y do obiektu. Jeśli masz inne niż jednolite skalę (skala, która różni się w każdym kierunku) tylko Mesh Collider można dopasować całkowicie.
- Jeśli przenosisz obiektu poprzez jego składnik Transform, ale chcesz otrzymywać wiadomości o kolizji/wyzwalaniu, możesz dołączyć Rigidbody do obiektu, który jest w ruchu.
- Jeśli tworzysz wybuch, może być bardzo skuteczne, dodanie rigidbody z dużą ilością drag i sphere collider do niego, aby przesunąć się trochę od ściany, którą trafi.

Zaawansowane

Kombinacja Collider-ów

Istnieje wiele różnych kombinacji kolizji, które mogą zajść w Unity. Każda gra jest wyjątkowa, i w różnych kombinacjach może działać lepiej dla różnego rodzaju gry. Jeśli korzystasz z fizyki w grze, to będzie bardzo pomocne, aby zrozumieć podstawowe typy Collider-ów, ich typowych zastosowań i ich wzajemne powiązania z innymi typami obiektów.

Statyczny Collider

Są GameObiekty, które nie mają dołączonego Rigidbody, ale mają dołączone Collider-y. Obiekty te powinny pozostać na swoim miejscu lub przenosić się niewiele. Doskonale dopasowują się do geometrii. Nie ruszają się jeśli Rigidbody zderza się z nimi.

Rigidbody Collider

To GameObiekt który zawiera zarówno Rigidbody jak i Collider. Dotyczą one całkowicie silnika fizyki poprzez siły ze skryptów i kolizje. Może zderzać się z GameObiektami, zawierającymi tylko Collider. Prawdopodobnie będą to podstawowe rodzaje Collider-ów w grze, które korzystają z fizyki.

Kinematic Rigidbody Collider

Ten GameObiekt zawiera Collider i Rigidbody, która oznaczony jest jako **Is Kinematic**. Aby przesunąć/przenieść ten GameObiekt, modyfikujesz jego **Transforms**, zamiast stosowania siły. Są one podobne do statycznego Collider-a, ale będzie działać lepiej, kiedy chcesz często przemieszczać Collider. Istnieje kilka innych wyspecjalizowanych scenariuszy korzystania z tego GameObiektu.

Ten obiekt może służyć do okoliczności, w których chcesz normalnie statyczny Collider wysłać wyzwalacz. Ponieważ wyzwalacz musi mieć dołączone Rigidbody, należy dodać Rigidbody, następnie włączyć **Is Kinematic**. To uniemożliwia przejście obiektu do wpływu fizyki pozwalając otrzymać wyzwolenie zdarzenia, kiedy chcesz.

Kinematic Rigidbody łatwo można włączać i wyłączać. To jest dobre dla tworzenia Ragdoll, gdy normalnie ma postać po animacji, a następnie przekształcić ragdoll po kolizji występującej spowodowanej przez eksplozję, czy cokolwiek innego, co wybrałeś. Gdy to się dzieje, po prostu włącz wszystkie swoje Kinematic Rigidbody w normalne Rigidbody za pomocą skryptów.

Jeśli masz Rigidbody które się nie poruszają przez jakiś czas, będzie „zasypiać”. Oznacza to, że one nie oblicza się podczas aktualizacji fizyki ponieważ nigdzie nie przenosi się. Jeśli przeniesiesz z Kinematic Rigidbody na normalne Rigidbody, które są w stanie spoczynku, śpiące Rigidbody „obudzi się” i poprawnie obliczy ponownie w aktualizacji fizyki. Więc jeśli masz dużo statycznych Collider-ów, że chcesz je poruszać i mają na nie inne obiekt wpadać poprawnie, użyj Kinematic Rigidbody Collider.

Kolizje działają macierzowo

W zależności od konfiguracji dwóch kolizji obiektów występuje wiele różnych działań. Tabela poniżej przedstawia to czego można się spodziewać od kolizji dwóch obiektów, oparte na składnikach, które są dołączone do nich. Niektóre kombinacje tylko spowodują jeden z dwóch obiektów po przez zderzenia więc pamiętaj standardową zasadę - fizyka nie zostanie za symulowana na obiektach które nie mają dołączonego Rigidbody.

Występuje wykrywanie kolizji i wiadomości są wysyłane po kolizji						
	Static Collider	Rigidbody Collider	Kinematic Rigidbody Collider	Static Trigger Collider	Rigidbody Trigger Collider	Kinematic Trigger Co
Static Collider		Y				
Rigidbody Collider	Y	Y	Y			
Kinematic Rigidbody Collider		Y				
Static Trigger Collider						
Rigidbody Trigger Collider						
Kinematic Rigidbody Trigger Collider						
Trigger wiadomości są wysyłane po kolizji						
	Static Collider	Rigidbody Collider	Kinematic Rigidbody Collider	Static Trigger Collider	Rigidbody Trigger Collider	Kinematic Trigger Co
Static Collider					Y	Y
Rigidbody Collider				Y	Y	Y
Kinematic Rigidbody Collider				Y	Y	Y
Static Trigger Collider		Y	Y		Y	Y
Rigidbody Trigger Collider	Y	Y	Y	Y	Y	Y
Kinematic Rigidbody Trigger Collider	Y	Y	Y	Y	Y	Y

Layer-Based Collision Detection

W Unity 3.x wprowadzono coś co nazywa się **Layer-Based Collision Detection**, a teraz selektywnie GameObiekty zderzają się w określonych warstwach, które są dołączone do Unity. Aby uzyskać więcej informacji kliknij [tutaj](#).

Box Collider

Box Collider podstawowym prymitywnym koliderem o kształcie sześcianu.

stos Box Collider-ów

Właściwości

- **Is Trigger** - Włączenie tego jest używane do wyzwalania zdarzenia i jest ignorowany przez silnik fizyki.
- **Material** - Odniesienie do [Fizycznego Materiału](#), który określa jak ten kolider współdziała z innymi.
- **Center** - Pozycja kolidera w lokalnej przestrzeni obiektu.
- **Size** - rozmiar kolidera w kierunkach X,Y,Z

Szczegóły

Box Collider może być zmieniany w różne kształty prostokątne, rąby. To świetna sprawa dla drzwi, ściany, platform, itp. Jest również skuteczny jako tors w ragdoll lub kadłub samochodu. Oczywiście działa idealnie na pudła i skrzynie również!

Standardowy Box Collider

Colliders współpracuje z Rigidbody pobudza fizykę w Unity do życia. Rigidbody dopuszcza kontrolę przez fizykę, Collider-y umożliwiają obiektom zderzanie się ze sobą. Collider musi zostać dodany do obiektu z Rigidbody. Collider nie koniecznie musi być dołączany do Rigidbody, ale Rigidbody musi być dołączone w celu aby obiekt przesuwiał się w wyniku kolizji.

W przypadku wystąpienia kolizji między dwoma Collider-ami i jeśli przynajmniej jeden z nich ma dołączony Rigidbody, **trzy wiadomości kolizji** są wysyłane do obiektów dołączonych do nich. Te zdarzenia mogą być obsługiwane w skryptach i pozwalają tworzyć unikalne zachowania z lub bez korzystania z wbudowanego silnika NVIDIA PhysX.

Trigger

Alternatywny sposób przy użyciu Collider-ów jest aby oznaczyć je jako **Trigger**, wystarczy zaznaczyć pole właściwość **Is Trigger** w Inspektorze. Trigger-y skutecznie są ignorowane przez silnik fizyki i ma unikalny zestaw **trzech wiadomości kolizji** które są wysyłane, gdy występuje kolizja z Trigger-ami. Wyzwalacze są przydatne do uruchomienia innych wydarzeń w grze, jak przerywniki, otwierające się automatycznie drzwi, wyświetlanie wiadomości poradnika, itp. Użyj wyobraźni!

Należy pamiętać, aby doszło do zderzenia, gdy zderzają się dwa wyzwalacze, jeden z nich musi zawierać również Rigidbody. Trigger zderzają się z normalnie z Collider-ami jeden z nich musi mieć dołączone Rigidbody. Zobacz szczegółowy schemat różnego rodzaju kolizji, działań macieży kolizji w sekcji **Zaawansowane** poniżej.

Tarcie i odbijanie

Tarcie, odbijanie i miękkość są zdefiniowane w **Fizycznych Materiałach, Standardowe Assety** zawierają najczęściej fizyczne materiały. Aby użyć jednego z nich kliknij na menu rozwijalne Physics Material i wybierz jeden, np. Ice. Można, także **tworzyć** własne materiały fizyczne i dostosować wszystkie wartości tarcia.

Mesh Collider

Mesh Collider tak jak **Mesh Asset** buduje się kolider oparty na tej siatce. Mesh Collider, które są oznaczone jako **Convex** mogą się zderzyć z innymi Mesh Collider-ami.

Mesh Collider działa z poziomą geometrią

Właściwości

- **Is Trigger** - jeśli włączone, Collider jest używany do wyzwalania zdarzenia i jest ignorowany przez silnik fizyki.
- **Material** - Odniesienie do **Fizycznego Materiału**, który określa, jak ten Collider współdziała z innymi.
- **Mesh** - odniesienie do siatki kolizji.
- **Smooth Sphere Collisions** - Gdy ta opcja jest włączona kolizje siatki są wygładzone. Powinno to umożliwić wygładzenie powierzchni np. na pofałdowanym terenie bez twardych krawędzi, aby toczenie się kuli było gładkie.
- **Convex** - Gdy włączone Mesh Collider zderza się z innymi Mesh Collider. Convex Mesh Colliders jest ograniczone do 255 trójkątów.

Szczegóły

Mesh Collider buduje źródło kolizji z siatki dołączonej do GameObiektu i czyta właściwości dołączonego Transform i skaluje się poprawnie do rozmiaru obiektu.

Mesh Collider wykorzystuje modelem. Jeśli obiekt zderza się z siatką, która jest odzwierciedlona graficznie to również nie koliduje z nim fizycznie.

Istnieją pewne ograniczenia przy użyciu Mesh Collider. Zwykle dwa Mesh Collider-y nie zderzają się ze sobą. Wszystkie Mesh Collider-y mogą się zderzyć z jakichkolwiek prymitywnym Collider-em. Jeśli twoja siatka jest oznaczona jako **Convex** wtedy może zderzać się z innymi Mesh Collider-ami.

Colliders współpracuje z Rigidbody pobudza fizykę w Unity do życia. Rigidbody dopuszcza kontrolę przez fizykę, Collider-y umożliwiają obiektom zderzanie się ze sobą. Collider musi zostać dodany do obiektu z Rigidbody. Collider nie necessarily musi być dołączany do Rigidbody, ale Rigidbody musi być dołączone w celu aby obiekt przesuwiał się w wyniku kolizji.

W przypadku wystąpienia kolizji między dwoma Collider-ami i jeśli przynajmniej jeden z nich ma dołączony Rigidbody, **trzy wiadomości kolizji** są wysyłane do obiektów dołączonych do nich. Te zdarzenia mogą być obsługiwane w skryptach i pozwalają tworzyć unikalne zachowania z lub bez korzystania z wbudowanego silnika NVIDIA PhysX.

Trigger

Alternatywny sposób przy użyciu Collider-ów jest aby oznaczyć je jako **Trigger**, wystarczy zaznaczyć pole właściwość **Is Trigger** w Inspektorze. Trigger-y skutecznie są ignorowane przez silnik fizyki i ma unikalny zestaw **trzech wiadomości kolizji** które są wysyłane, gdy występuje kolizja z Trigger-ami. Wyzwalacze są przydatne do uruchomienia innych wydarzeń w grze, jak przerywniki, otwierające się automatycznie drzwi, wyświetlanie wiadomości poradnika, itp. Użyj wyobraźni!

Należy pamiętać, aby doszło do zderzenia, gdy zderzają się dwa wyzwalacze, jeden z nich musi zawierać również Rigidbody. Trigger zderzają się z normalnie z Collider-ami jeden z nich musi mieć dołączone Rigidbody. Zobacz szczegółowy schemat różnego rodzaju kolizji, działań macieży kolizji w sekcji **Zaawansowane** poniżej.

Tarcie i odbijanie

Tarcie, odbijanie i miękkość są zdefiniowane w **Fizycznych Materiałach, Standardowe Assety** zawierają najczęściej fizyczne materiały. Aby użyć jednego z nich kliknij na menu rozwijalne Physics Material i wybierz jeden, np. Ice. Można, także **tworzyć** własne materiały fizyczne i dostosować wszystkie wartości tarcia.

Wskazówki

- Mesh Collider nie zderzają się ze sobą, chyba że są one oznaczone jako **Convex**. W związku z tym że są najbardziej przydatne dla obiektów tła, takich jak geometria środowiska.
- Convex Mesh Colliders musi być mniejsze niż 255 trójkątów.
- Prymitywne kolidery są mniej kosztowne dla obiektów jeśli chodzi o wydajność kiedy są pod kontrolą fizyki.
- Po dołączeniu Mesh Collider do GameObiektu jego właściwość Mesh będzie domyślna renderowana siatka. Można to zmienić przez przypisywanie różnych siatek.
- Aby dodać wiele Collider-ów dla obiektu, utwórz dziecko GameObiekt i dołącz kolidery do każdego z nich. Dzięki temu każdym Collider-em manipulujesz niezależnie.

- Można spojrzeć na gizmo w widok sceny, aby zobaczyć, jak Collider jest przeliczany na obiekcie.
- Jeśli przenosisz obiektu poprzez jego składnik Transform, ale chcesz otrzymywać wiadomości o kolizji/wyzwalania, możesz dołączyć Rigidbody do obiektu, który jest w ruchu.

Fizyczne Materiały

Physic Material jest używany w obiektach do regulacji skutków tarcia/odbicia gdy dojdzie do kolizji obiektów.

Aby stworzyć Physic Material wybierz **Assets**→**Create**→**Physics Material** z górnego menu. Następnie przeciągnij fizyczny materiał na Collider w Widoku Projektu.

Inspektor Fizyczne Materiałów

Właściwości

- **Dynamic Friction** - Tarcie stosowane gdy już się porusza. Zazwyczaj wartość od 0 do 1. Wartość zero jest jak lód, wartość 1 pozwoli się zatrzymać bardzo szybko, chyba że duża siła lub ciężar popycha obiekt.
- **Static Friction** - stosowane tarcie gdy obiekt jest nadal na powierzchni. Zazwyczaj wartość od 0 do 1. Wartość zero jest jak lód, z wartością 1 będzie bardzo trudno uzyskać obiekt poruszający się.
- **Bounciness** - Jak odbita jest powierzchnia? Wartość 0 nie zostanie odbita. Wartość 1 będzie odbijać bez utraty energii.
- **Friction Combine Mode** - Jak tarcie dwóch zderzających się obiektów jest połączone.
- **Average** - Dwie wartości tarcia są uśredniane.
- **Min** - Najmniejsza z dwóch wartości jest używana.
- **Max** - Największa z dwóch wartości jest używana.
- **Multiply** - Wartości tarcia są mnożone wzajemnie.
- **Bounce Combine** - Jak odbijanie się dwóch obiektów zderzających się łączy. Ma te same tryby jak Friction Combine Mode.
- **Friction Direction 2** - Kierunku anizotropii. Anizotropowe tarcie jest włączone jeśli ten kierunek nie jest równy zero. Dynamic Friction 2 i Static Friction 2 będzie stosowany przy Friction Direction 2.
- **Dynamic Friction 2** - Jeśli anizotropowe tarcie jest włączona, Dynamic Friction 2 będzie stosowany przy Friction Direction 2.
- **Static Friction 2** - Jeśli włączone jest anizotropowe tarcie, Static Friction 2 będzie stosowany przy Friction Direction 2.

Szczegóły

Tarcie zapobiega zsuwaniu się z siebie powierzchni. Wartość ta ma kluczowe znaczenie podczas próby układania obiektów w stosy. Tarcie występuje w dwóch formach: dynamiczne i statyczne. **Static Friction** jest używane do przedmiotów leżących nieruchomo. Zapobiega to temu że obiekty zaczynają się ruszać. Jeśli duża siła jest przyłożona do obiektu wtedy zacznie się ruszać. W tym momencie **Dynamic Friction** będzie wchodzić w grę. **Dynamic Friction** będzie teraz próbować zwolnić obiekt z innym.

Wskazówki

Nie próbuj używać standardowego fizycznego materiału dla głównego bohatera. Wykonaj jeden doskonale dostosowany.

Hinge Joint

Hinge Joint grupuje razem dwa Rigidbody, ograniczające im poruszanie się jak są one połączone zawiasem. Nadaje się do drzwi, ale może również służyć do modelu łańcucha, wahadełek itd.

Inspektor Hinge Joint

Właściwości

- **Connected Body** - Opcjonalne odniesienie do Rigidbody, że staw jest uzależniony. Jeśli nie zestaw wspólne łączy się ze światem.
- **Anchor** - Położenie osi, wokół której ciało się huśta. Położenie jest określone w lokalnej przestrzeni.
- **Axis** - Kierunki osi, wokół którego ciało się huśta. Kierunek jest zdefiniowany w lokalnej przestrzeni.
- **Use Spring** - Sprawia sprężynowanie Rigidbody w porównaniu do jego osiągniętego dla danego kąta połączonego ciała.
- **Spring** - Właściwości sprężynowania, które są używane, jeśli **Use Spring** jest włączona.
- **Spring** - Siła obiekt potrzeba aby wrócić/przejsć do pozycji.
- **Damper** - Im wyższa ta wartość, tym bardziej obiekt spowolni.
- **Target Position** - Docelowy kąt skoku. sprężynowanie ciągnie się w kierunku tego kąta, mierzone w stopniach.
- **Use Motor** - Ruch sprawia, że obiekt obracać się wokół.
- **Motor** - Właściwości ruchu, które są używane, jeśli **Use Motor** jest włączona.
- **Target Velocity** - Prędkość obiektu którą próbuje osiągnąć.
- **Force** - Siła którą stosuje się w celu osiągnięcia prędkości.
- **Free Spin** - Jeśli włączone, ruch nie jest wykorzystywany do hamowania obracania, tylko przyspieszenia.
- **Use Limits** - Jeśli włączone, kąt zależeć będą od ograniczeń w wartości **Min** i **Max**.
- **Limits** - Właściwości ograniczenia, które są używane, jeśli **Use Limits** jest włączone.
- **Min** - Najniższy kąt obrotu w jaki może przejść.
- **Max** - Najwyższy kąt obrotu w jaki może przejść.
- **Min Bounce** - Minimalna wartość ile razy obiekt musi się odbić kiedy w coś trafi aby się zatrzymać.
- **Max Bounce** - Maksymalna wartość ile razy obiekt musi się odbić kiedy w coś trafi aby się zatrzymać.
- **Break Force** - Siła które należy zastosować do tego aby złamać.
- **Break Torque** - Moment obrotowy który należy zastosować do tego aby złamać.

Szczegóły

Jeden Hinge Joint powinien być stosowany do GameObiektu. Zawias obróci się w punkcie określonym przez właściwość **Anchor**, właściwości poruszania się określa **Axis**. Nie trzeba przypisywać GameObiektowi wspólnej własności **Connected Body**. Tylko należy przypisać GameObiektowi z właściwością **Connected Body** jeśli chcesz wspólny **Transform** zależny od **Transform** dołączonego obiektu.

Zastanów się, jak działa zawias drzwi. **Axis** w tym przypadku jest, dodatkowo wzdłuż osi Y. **Anchor** znajduje się gdzieś na skrzyżowaniu między drzwiami i ścianą. Nie trzeba przypisać ścianie **Connected Body**, ponieważ staw będzie podłączony do świata domyślnie.

Teraz pomyśl o zawias drzwi przesłony. **Axis** przesłony drzwi będzie bokiem, dodatkowo wzdłuż względem osi X. Główne drzwi powinny być przypisane jako **Connected Body** więc zawias przesłony drzwi zależy od Rigidbody głównych drzwi.

Łańcuchy

Wiele Hinge Joints można również nawlekać razem, aby stworzyć łańcuch. Dodaj staw każdego ogniw w łańcuchu i dołącz następne połączenie jako **Connected Body**.

Wskazówki

- Nie trzeba przypisywać **Connected Body** do twojego Joint pracować.
- Właściwości **Spring**, **Motor** i **Limits** umożliwiają dostosowanie wspólnych zachowań.

Spring Joint

Spring Joint grupuje razem dwa **Rigidbody** ograniczając je wrzucając połączenie sprężyną .

Inspektor Spring Joint

Właściwości

- **Connected Body** - Opcjonalne odniesienie do Rigidbody, gdy staw jest uzależniony od innego.
- **Anchor** - Pozycji obiektu w lokalnej przestrzeni definiuje się w środku stawu. Nie jest to punkt w który obiekt będzie kierować się w kierunku do.
- **X** - Pozycja stawu w lokalnym centrum wzdłuż osi X.
- **Y** - Pozycja stawu w lokalnym centrum wzdłuż osi Y.
- **Z** - Pozycja stawu w lokalnym centrum wzdłuż osi Z.
- **Spring** - Siła sprężyny.
- **Damper** - Liczba przy której sprężyna jest zmniejszana gdy jest aktywna.
- **Min Distance** - Odległość większa niż ta nie spowoduje że sprężyna się uaktywni.
- **Max Distance** - Odległość mniejsza niż ta nie spowoduje że sprężyna się uaktywni.
- **Break Force** - Siła, które należy zastosować do tego wspólnego złamania.
- **Break Torque** - Moment obrotowy, który należy zastosować do tego wspólnego złamania.

Szczegóły

Spring Joints pozwalają Rigidbody GameObiektu być przyciąganym do pozycji określonej „miejsce docelowe”. Pozycja ta będzie kolejny Rigidbody GameObiektu lub świat. Jak GameObiekt podróżuje dalej od pozycji tego „punktu docelowego”, sprężyna dotyczy siły, które będą

wyciągać ją z powrotem do jego pierwotnej pozycji „miejsce docelowego”. To daje efekt bardzo podobny do gumki lub procy.

„Miejsce docelowe” sprężyny zależy od położenia **Anchor** do **Connected Body** (lub świata) po utworzeniu wspólnej sprężyny lub jeżeli wprowadzony zostanie tryb Play mode. To czyni wspólne sprężyny bardzo skutecznymi na tworzenie sprężynujących znaków lub obiektów w edytorze, ale jest trudniej stworzyć wypychania/ściągnięcia zachowań sprężyn w czasie wykonywania za pomocą skryptów. Jeśli chcesz kontrolować przede wszystkim pozycję GameObiektu za pomocą wspólnej sprężyny, to najlepiej utworzyć pusty GameObiekt z Rigidbody i **Connected Rigidbody** zawiasami obiektów. Następnie w skrypcie można zmienić pozycję **Connected Rigidbody** i zobaczyć twoją sprężynę poruszającą się w sposób, którego oczekujesz.

Connected Rigidbody

Nie musisz używać **Connected Rigidbody** do pracy twojego zawiasu. Ogólnie rzecz biorąc należy używać tylko jeden jeśli pozycja twojego obiektu i/lub obrót jest zależy od innego. Jeśli nie jest Connected Rigidbody, twój sprężyna połączy się ze światem.

Spring i Damper

Spring jest siłą, która zwraca obiekt do powrotu do pozycji „miejsca docelowego”. Jeśli jest to 0, to nie ma siły, która będzie działać na obiekt i to będzie zachowywać się, jakby Spring Joint nie było dołączone w ogóle.

Damper jest to opór który napotka siłą sprężyny. Im niższe tym bardziej sprężysty obiekt będzie. Jak zwiększa się Damper zmniejsza się ilość odbić, spowodowane przez wspólny Joint.

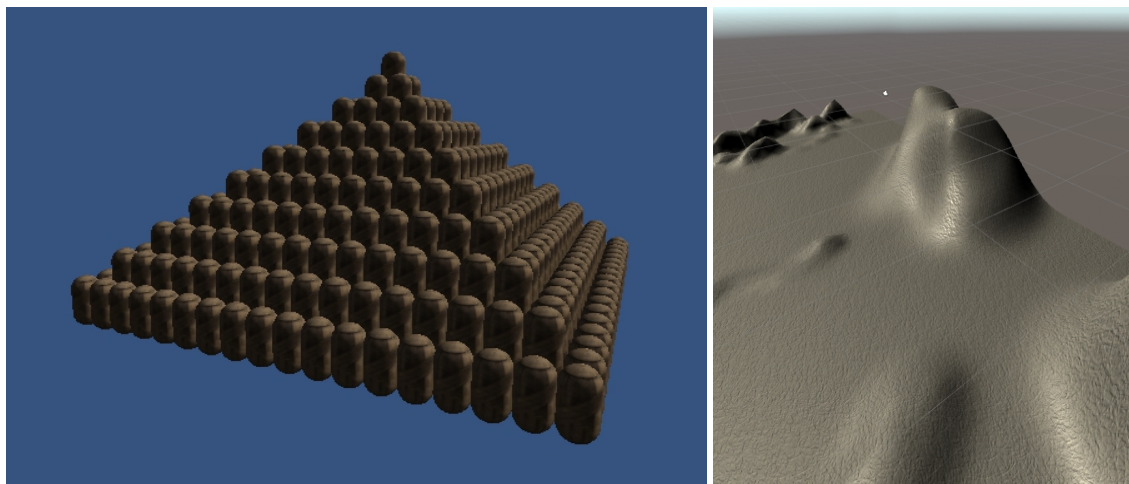
Min i Max Distance

Jeśli położenie obiektu mieści się pomiędzy **Min i Max Distance** następstwem tego będzie że staw nie zostanie zastosowany do obiektu. Pozycja musi zostać przesunięta poza te wartości do aktywacji stawu.

Wskazówki

- Nie potrzeba dodawać Connected Body aby twój Joint pracował.
- Zestaw idealny pozycje obiektów Jointed w edytorze przed wejściem do trybu odtwarzania Play mode.
- Spring Joints wymaga dołączone do obiektu Rigidbody

Warianty zadania



Program 1. Narysować piramidę z użyciem elementów podstawowych wg wariantu

1. kula
2. sześcián
3. cylinder

Ilość poziomów piramidy jest 10.

Piramida jest położona na powierzchni ziemi (obiekt Terrain) pokrytą górami. Użyć materiały dla powierzchni ziemi (paczka Tangy)

1. grass
2. moon
3. concrete

Połączyć elementy podstawowe piramidy sprężynami (Spring Joint)

Opracować grę w której gracz za pomocą kuli chce rozbijać piramidę (używając Constant Force)

P.S. W przypadku sciany złożonej z sześciánów wzór kodu źródłowego

<http://www.eecs.tufts.edu/~mshah08/comp/150GD/lecture2/lecture2.zip>,

wzór gry <http://www.eecs.tufts.edu/~mshah08/comp/150GD/lecture2/Builds/Builds.html>