

Desenvolvimento com Motores de Jogos II

Aula 13 - Colliders

Apresentação

Saudações, pessoal! Estamos na aula 13 de Motores de Jogos II, nos aproximando do final da disciplina. Mas nada de tristeza, pois ainda não completamos o trajeto, vamos lá! Nesta aula, estudaremos os tipos de colisores, chamados de **Colliders**, disponíveis no Unity, assim como analisaremos o tipo de comportamento que cada um adiciona aos objetos de uma cena.

Veremos em detalhes cada colisor, aprenderemos a adicioná-los em objetos e entender qual a relação dos **Colliders** com o comportamento dos movimentos e interações físicas no Unity.

Na próxima aula, vamos colocar a “mão na massa” e utilizar o que já aprendemos, conto com vocês!

Objetivos

Ao final desta aula, você deverá ser capaz de:

- Compreender o conceito de Colliders (Colisores) no Unity;
- Identificar os tipos de Colliders e seus comportamentos;
- Adicionar Colliders em modelos 3D.

Colliders

Colliders são geometrias especiais que podem ser adicionadas a qualquer GameObject no Unity e adicionam o recurso de colisão física nesses objetos. São componentes invisíveis e não precisam ter o mesmo formato que o objeto ao qual ele está adicionado, ou seja, um objeto no formato de uma caneca de café (como a da aula passada), por exemplo, pode ter um **Collider** no formato de um simples cubo (Box Collider), fazendo com que ele, para todos os efeitos de interações físicas, comporte-se como um cubo. Isso parece estranho, mas na verdade é uma ótima solução para questões de performance do seu jogo, já que calcular interações físicas com **Colliders** simples (como o Box Collider) é mais rápido do que considerar todos os detalhes de um objeto como uma caneca no processamento da física do jogo.

Os **Colliders** mais simples são chamados de **primitive colliders** (Colisores Primitivos), e em 3D os disponíveis são:

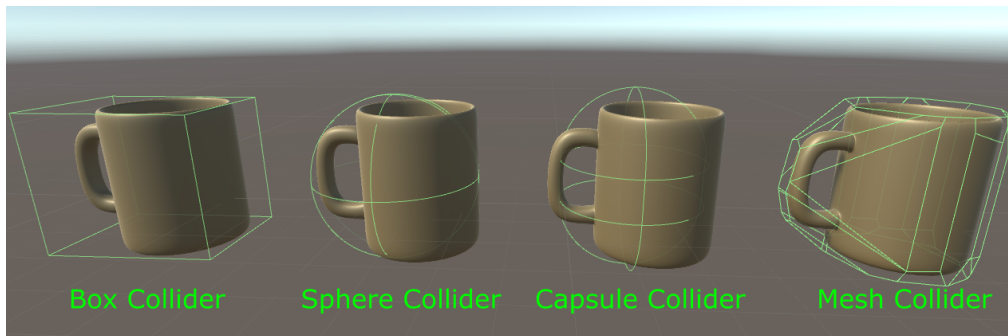
- **Box Collider:** Colisor em formato de uma caixa (com 6 faces).
- **Sphere Collider:** Colisor em formato de uma esfera.
- **Capsule Collider:** Colisor em formato de uma cápsula.

Outros tipos comuns de **Colliders**, entretanto com menor performance (maior uso de processador durante o jogo), são:

- **Mesh Collider:** Colisor mais genérico que pode ter qualquer formato geométrico, bastando associá-lo a um Mesh 3D qualquer.
- **Wheel Collider:** Colisor especial utilizado para rodas de veículos.
- **Terrain Collider:** Colisor especial para uso com elevação de terrenos.

Na **Figura 1** você pode ver os tipos de colliders (menos o Wheel Collider) aplicados no modelo 3D utilizado no nosso projeto passado (caneca). Observe que ao selecionar os objetos o seu **Collider** é exibido na Scene View como uma geometria de linhas verdes ao redor do objeto.

Figura 01 - Quatro tipos de Colliders aplicados em um modelo 3D.



Fonte: Elaborado pelo autor (2017).

No Unity não existe um Collider primitivo com o formato de um cilindro. Assim, se você adicionar um GameObject do tipo Cylinder o componente colisor que vem associado com esse objeto é um Capsule Collider. Esse tipo de deficiência é causado pelo fato do Unity utilizar um motor de física chamado PhysX que não suporta colisores em formato de cilindro. No entanto isso não impede que utilize um Mesh Collider nos seus objetos e escolha uma geometria em formato cilíndrico (ou qualquer outro formato) para ele. A única diferença é que você não estará se beneficiando das melhorias de performance por utilizar um primitive collider.

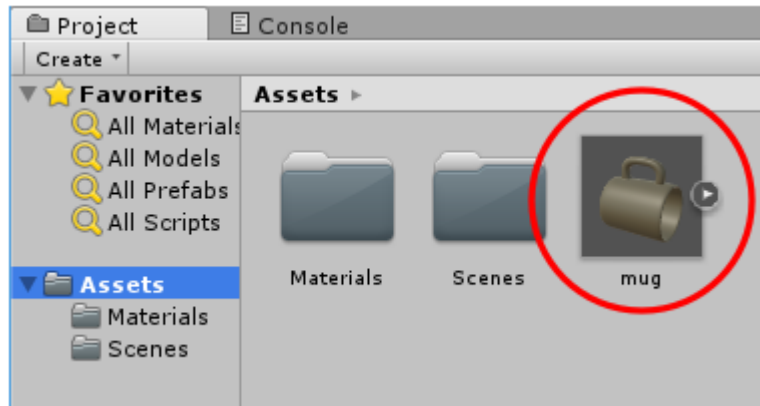
Adicionando Colliders em Objetos

Ao criar um objeto 3D primitivo no Unity (Cube, Sphere, etc.), ele já vem com um componente **Collider** associado, entretanto, se você importar um Mesh 3D no Unity, esse objeto não vem com um **Collider**, cabendo a você adicionar esse componente.

Para esta aula, utilizaremos o mesmo projeto em que adicionamos um modelo de uma caneca 3D no Unity, todavia você poderá utilizar qualquer outro modelo de sua preferência.

Lembrando que o nosso projeto já deve conter um modelo 3D de uma caneca (chamado “mug”) na pasta raiz dos assets, como mostra a Figura 2. O modelo utilizado foi baixado do link <https://www.turbosquid.com/3d-models/cup-max-free/513427>, então você precisa ter uma conta no TurboSquid (demonstrado na aula passada) para baixá-lo. Talvez esse modelo não esteja mais disponível quando estiver vendo esta aula, entretanto qualquer outro servirá para acompanhá-la.

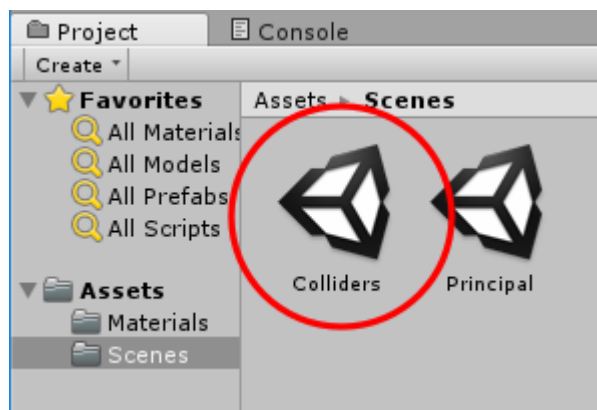
Figura 02 - Pasta do projeto com um modelo 3D (de uma caneca) criado.



Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

Crie uma nova cena e salve-a com o nome de “Colliders” na pasta Scenes, obtendo um resultado como o mostrado na **Figura 3**.

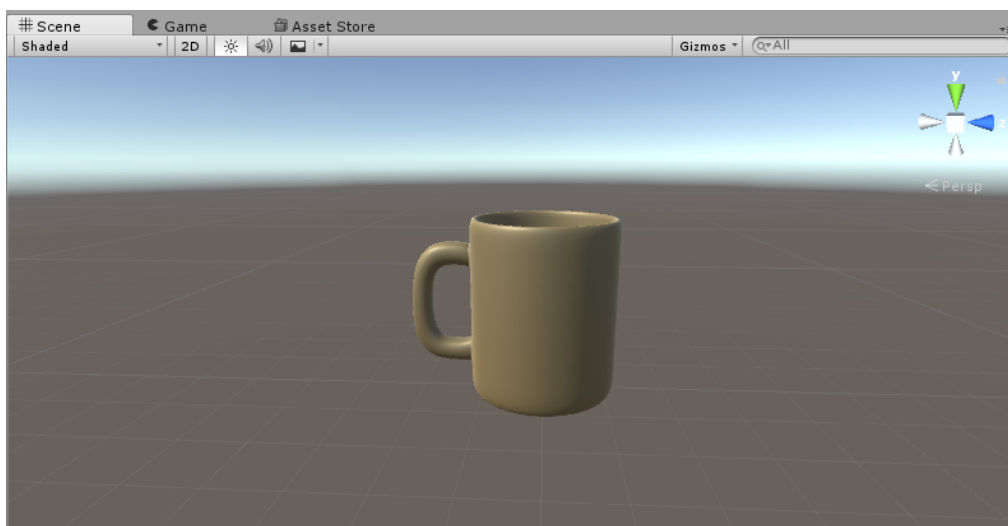
Figura 03 - Nova cena criada.



Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

Adicione na sua cena uma instância do modelo “mug” (a caneca), arrastando-o da pasta Assets para o Scene View e, logo em seguida, modifique sua posição para 0,0,0 colocando-o no centro da cena. Clique duas vezes no objeto “mug” criado no Hierarchy e você visualizará o scene view de forma mais clara, como na **Figura 4**.

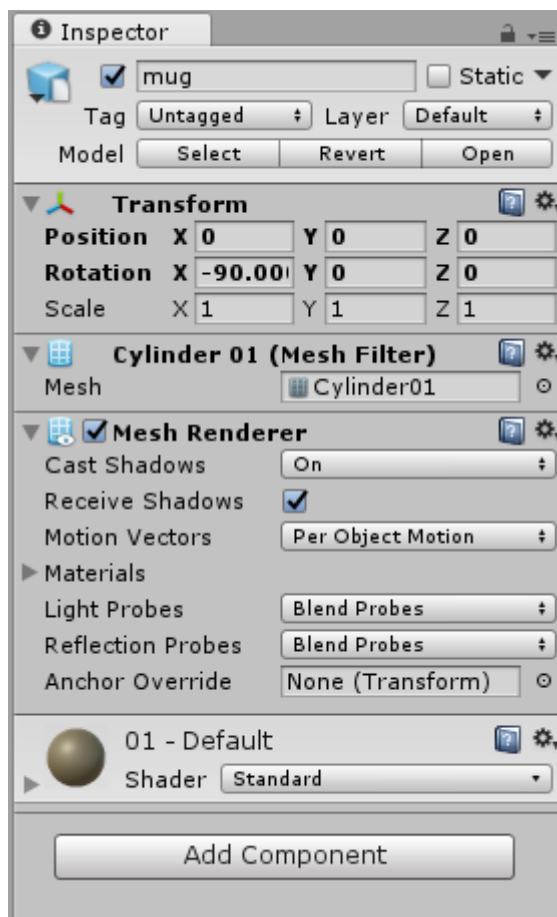
Figura 04 - Caneca no centro da cena.



Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

Clique no objeto “mug” no Hierarchy e modifique seu nome para “Caneca”. Logo em seguida, com a Caneca selecionada, observe o Inspector e veja que não existe nenhum componente de colisão. Perceba esse detalhe na **Figura 5**.

Figura 05 - Detalhes no Inspector do objeto Caneca sem nenhum colisor associado.



Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

Repare no Inspector que a Caneca só tem um Transform; um Mesh Filter – nesse caso chamado de Cylinder 01 (nome dado pelo modelador) – com o modelo 3D (Mesh) exibido; um Mesh Renderer com as configurações avançadas relativas ao modo como esse objeto será exibido (usamos os valores padrão); e um Material associado (01 - Default), o qual é o Material utilizado no objeto com o Standard Shader.

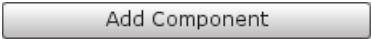
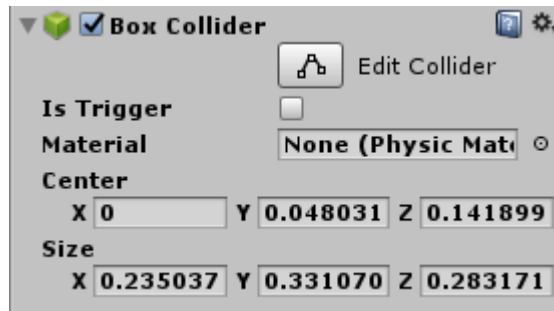
Clique agora no botão Add component  escolha a opção Physics -> Box Collider e um novo componente será adicionado na Caneca, como visto no Inspector e detalhado na **Figura 6**.

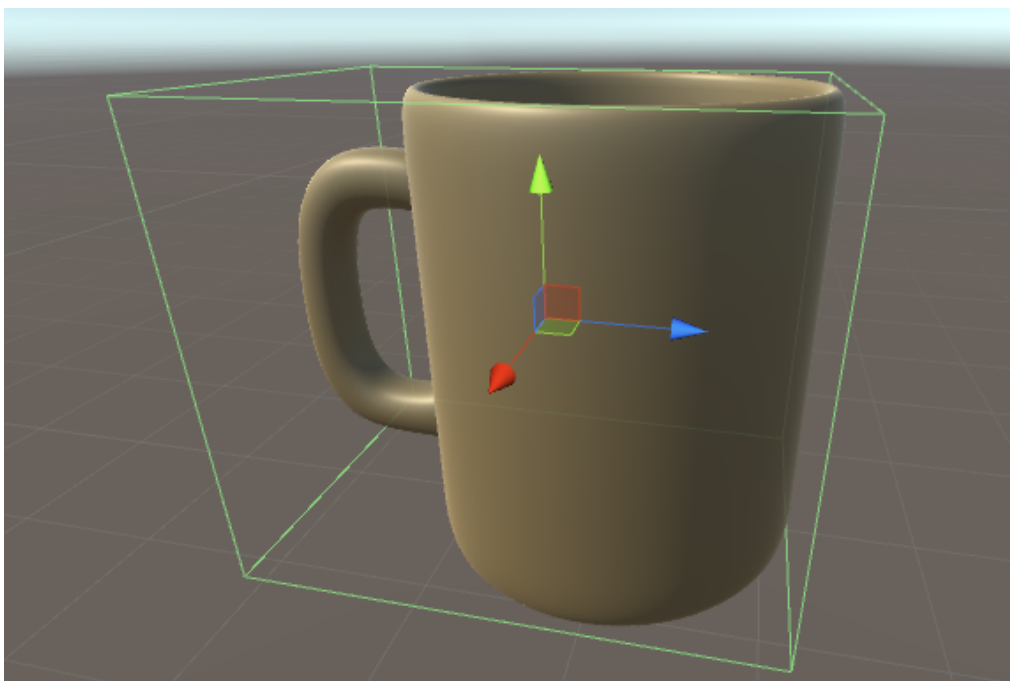
Figura 06 - Box Collider no Inspector da Caneca.



Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

Veja que o **Box Collider** já vem com vários valores configurados por padrão. Esses valores são relacionados a suas dimensões, ao caso de ele ser tipo “Is Trigger” e à sua posição relativa à Caneca. Por padrão o Unity cria um **Box Collider** que “envolve” todo o modelo 3D e é exibido também no Scene View com um contorno verde em formato de “caixa” quando o objeto está selecionado. Veja a **Figura 7**.

Figura 07 - Contorno do Box Collider adicionado na Caneca.



Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

Pronto, agora sua Caneca tem um colisor do tipo **Box Collider**. Isso significa que o Unity considera, para efeitos de interações físicas e colisões, sua Caneca como uma simples “caixa”, de modo a facilitar muito o cálculo de colisões do motor de física do Unity.

Atividade 01

1. Com a sua Caneca selecionada, modifique os valores do atributo “Center” do Box Collider adicionado e observe o que acontece com sua representação (contornos verdes) no Scene View.
2. Com a sua Caneca selecionada, modifique os valores do atributo “Size” do Box Collider adicionado e observe o que acontece com sua representação (contornos verdes) no Scene View.

Tipos de Colliders

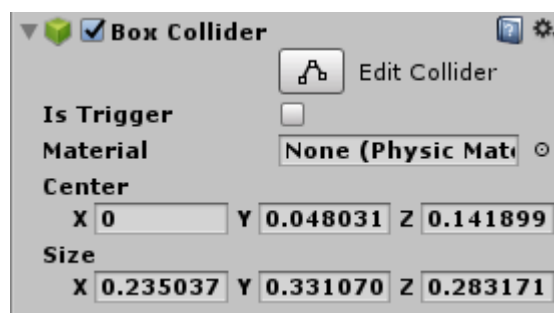
Como dito anteriormente, o Unity fornece alguns tipos primitivos de colisores (Box, Sphere e Capsule) assim como outros tipos não primitivos (Mesh, Wheel e Terrain). Nesta aula, estudaremos os tipos primitivos e, também, o tipo não primitivo **Mesh Collider**, devido a esses serem os mais populares e suficientes para a construção de vários tipos de jogos.

Box Collider

Um dos colisores primitivos do Unity é o **Box Collider**. Ele se trata de uma “caixa” que pode ser adicionada em qualquer objeto da cena.

O **Box Collider** tem quatro atributos que podem ser observados na **Figura 8**.

Figura 08 - Atributos do Box Collider



Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

Abaixo, segue a descrição dos quatro atributos do **Box Collider**:

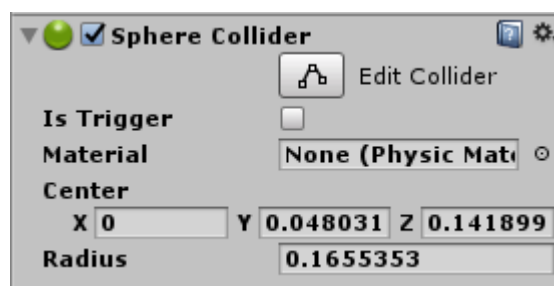
- **Is Trigger:** se estiver habilitado, o colisor é utilizado para “eventos de gatilho” (Triggering Events), ou seja, as reações físicas em uma colisão são ignoradas, mas o Unity as detecta, permitindo a você escrever um script que seja executado quando ela ocorrer.
- **Material:** permite a você escolher um Material Físico para esse Collider. Isso determina como ele interage com outros colisores. Não exploraremos em detalhes os Physics Materials nesta aula por se tratar de um assunto mais avançado, entretanto você pode estudar mais sobre eles verificando os links indicados na seção de Leitura Complementar.
- **Center:** a posição relativa (X,Y,Z) do centro do Box Collider em relação ao modelo.
- **Size:** o tamanho do Box Collider nos eixos X, Y e Z.

Para adicionar um **Box Collider** em um objeto, basta selecioná-lo e escolher no Inspector a opção Add Component -> Physics -> Box Collider, assim como foi visto anteriormente.

Sphere Collider

O **Sphere Collider** é um tipo de colisor primitivo que tem um formato de uma simples esfera. Para adicionar um **Sphere Collider**, o processo é similar ao dos outros colisores, bastando selecionar o objeto desejado e, no Inspector, clicar no botão Add Component escolhendo a opção Physics -> Sphere Collider. Você terá um componente como o mostrado na **Figura 9**.

Figura 09 - Atributos do Sphere Collider.



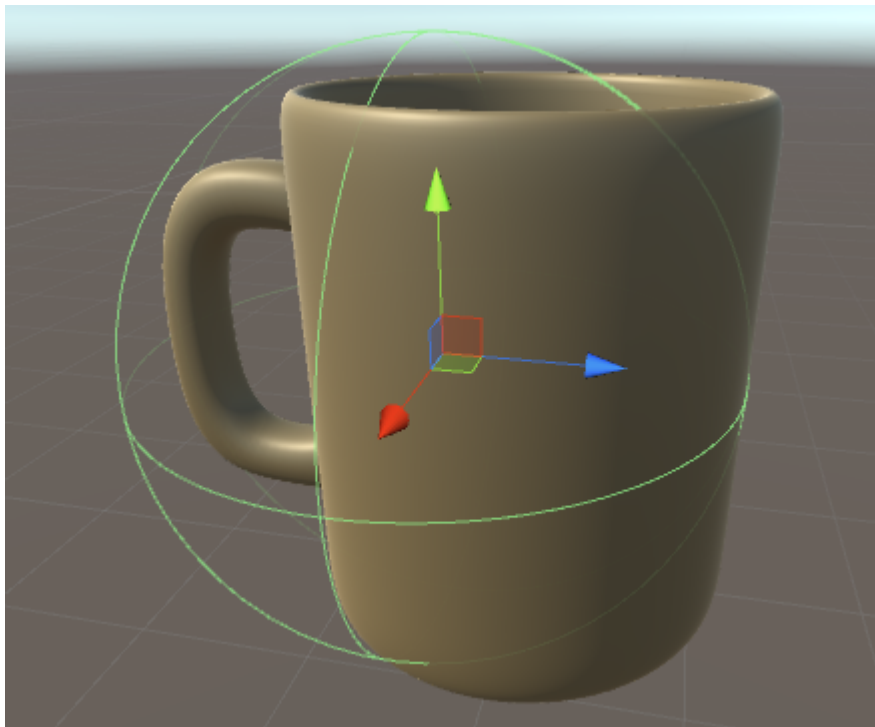
Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

Como você pode reparar, o Sphere Collider tem os seguintes atributos:

- **Is Trigger:** assim como no Box Collider, se habilitado, o colisor é utilizado para “eventos de gatilho”.
- **Material:** o Physics Material que será utilizado.
- **Center:** a posição relativa (X,Y,Z) do centro do Sphere Collider em relação ao modelo.
- **Radius:** o tamanho do raio da esfera.

Na **Figura 10**, você pode ver um objeto com o componente **Sphere Collider** adicionado.

Figura 10 - Sphere Collider em um objeto na Scene View.

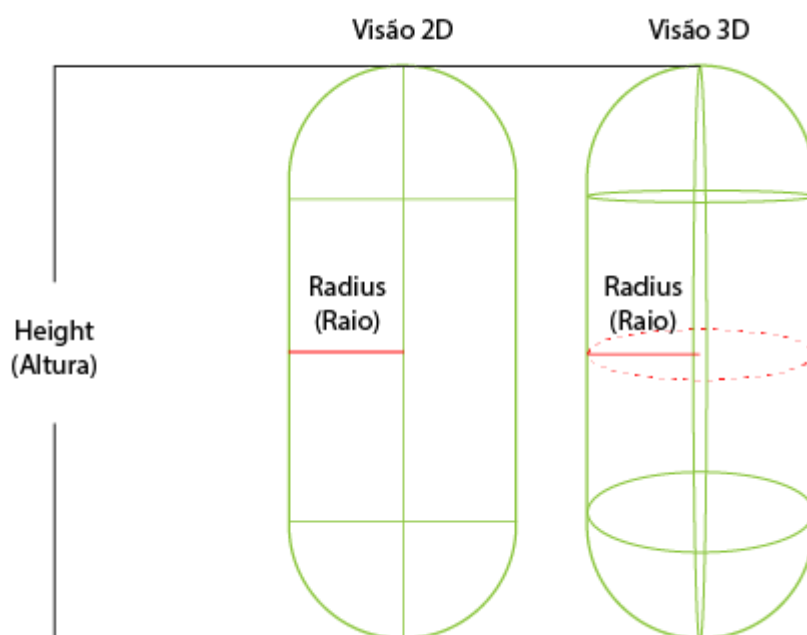


Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

Capsule Collider

O **Capsule Collider** é um colisor primitivo do Unity. Esse colisor tem um formato de uma cápsula, ou seja, algo como um cilindro com as pontas no formato de uma semiesfera, como é visto na **Figura 11**.

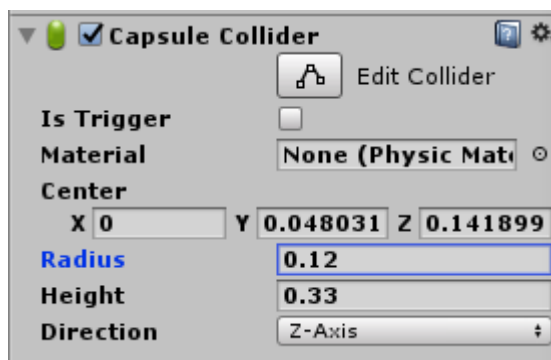
Figura 11 - Representação de um Capsule Collider.



Fonte: Elaborada pelo autor (2017)

Depois de adicionado, o **Capsule Collider** aparecerá como um componente do objeto. Veja a **Figura 12**.

Figura 12 - Capsule Collider no Inspector de um objeto.



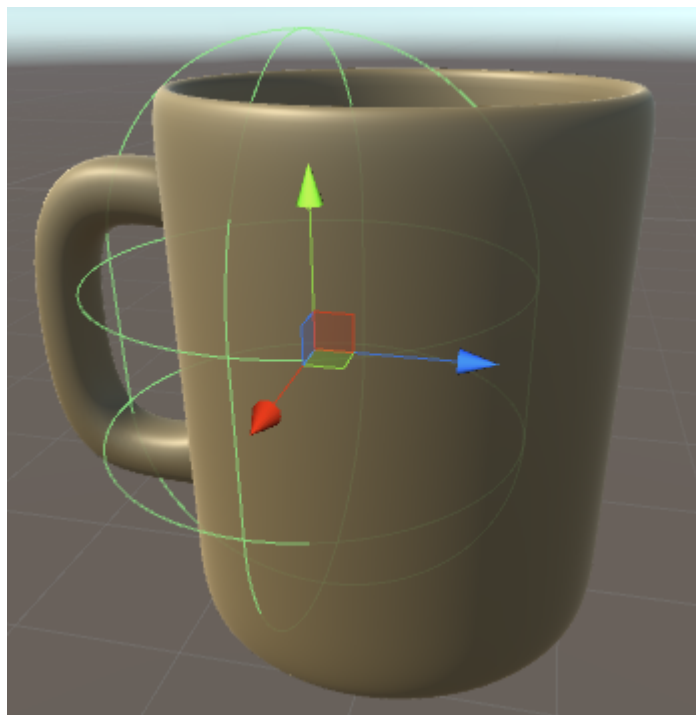
Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

Os atributos de um **Capsule Collider** são:

- **Is Trigger:** assim como nos outros Colliders, se habilitado, o colisor é utilizado para “eventos de gatilho”.
- **Material:** o Physics Material que será utilizado.
- **Center:** a posição relativa (X,Y,Z) do centro do Capsule Collider em relação ao modelo.
- **Radius:** o tamanho do raio da cápsula.
- **Direction:** em qual eixo (X, Y ou Z) o Capsule Collider está orientado. É necessária essa informação, pois ele não é um colisor simétrico.

Na **Figura 13**, temos um **Capsule Collider** adicionado no modelo 3D da Caneca, porém com a orientação modificada para o eixo Z (Z-Axis).

Figura 13 - Capsule Collider no modelo 3D da Caneca e orientação no Eixo Z.



Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

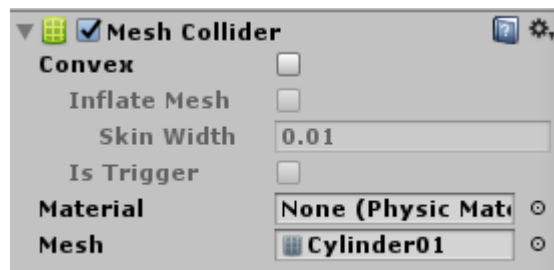
Mesh Collider

O **Mesh Collider**, diferente dos colisores primitivos (Box Collider, Sphere Collider e Capsule Collider), é um colisor especial que não tem um formato geométrico definido, ou seja, ele utiliza um modelo 3D previamente criado como base para o seu formato.

O **Mesh Collider** é muito útil quando você precisa criar um colisor mais complexo, o qual tenha sua própria geometria. Assim, as interações físicas com o objeto ao qual ele está adicionado serão muito mais realistas, entretanto isso terá um alto custo de performance se o modelo 3D utilizado para o colisor tiver muitas faces e vértices. Use colisores primitivos sempre que possível.

Para adicionar um **Mesh Collider**, selecione um objeto (a Caneca, por exemplo) e clique no botão Add Componente, logo em seguida, escolhendo a opção Physics -> Mesh Collider. Isso fará com que um componente **Mesh Collider** apareça no Inspector do objeto. Veja a **Figura 14**.

Figura 14 - Mesh Collider no inspector do objeto (Caneca).



Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

As propriedades do Mesh Collider são:

- **Convex:** se marcado, esse colisor interage com outros colisores que tenham Rigidbody (explicaremos melhor adiante).
- **Inflate Mesh:** se marcado, faz o Unity “inflar” o colisor gerado, de modo que ele fique um pouco maior.

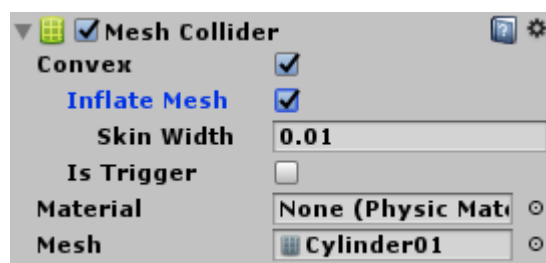
- **Skin Width:** define quanto maior o colisor marcado com Inflate Mesh será. O valor padrão de 0.01 é normalmente o suficiente.
- **Is Trigger:** assim como nos outros Colliders, se habilitado, o colisor é utilizado para “eventos de gatilho”.
- **Material:** o Physics Material que será utilizado.
- **Mesh:** refere-se ao modelo 3D que o Mesh Collider utilizará como referência para sua geometria. Por padrão, ele utiliza o mesmo modelo 3D do objeto ao qual ele está adicionado, entretanto você pode escolher outro modelo 3D (normalmente um modelo mais simplificado).

Logo depois de adicionar um **Mesh Collider**, pode ser que você não veja nenhum contorno verde no objeto representando sua geometria. Para isso, você precisa marcar a opção “Convex” e, em seguida, “Inflate Mesh” (normalmente opcional).

A opção Convex é muito importante, pois cria um colisor baseado no **Mesh** escolhido (por padrão, o do próprio objeto) e limitado a 255 faces. Isso faz com que, mesmo sendo um colisor de geometria personalizada, o Unity não tenha tanto trabalho calculando sua colisão com outros Colliders. Se essa opção não estiver marcada, o Mesh Collider não entrará em colisão com outros Mesh Colliders, portanto é importante marcá-la.

Marque as opções **Convex** e **Inflate Mesh**, conforme o exemplo da **Figura 15**.

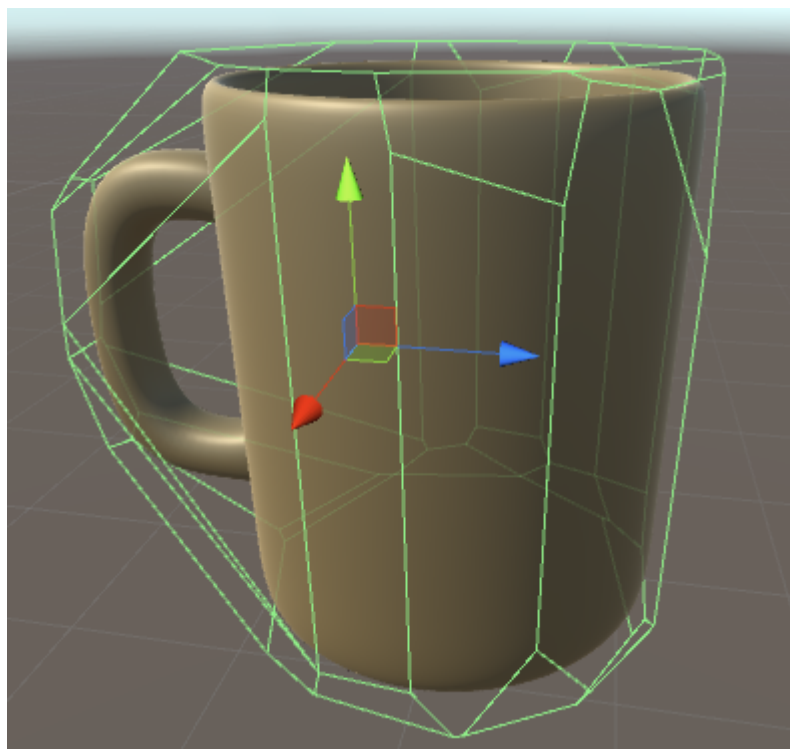
Figura 15 - Mesh Collider com os atributos Convex e Inflate marcados.



Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

Como já falado, o atributo **Mesh** já vem com o **Mesh** do próprio modelo selecionado (no caso da nossa Caneca, esse Mesh tem o nome de Cylinder01). Quando marcamos as opções **Convex** e **Inflate Mesh** (esta talvez não seja necessária), um colisor de geometria aproximado ao do escolhido no atributo Mesh será criado e você poderá visualizá-lo no Scene View, como mostra a **Figura 16**.

Figura 16 - Mesh Collider na caneca com as opções **Convex** e **Inflate** marcadas.



Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

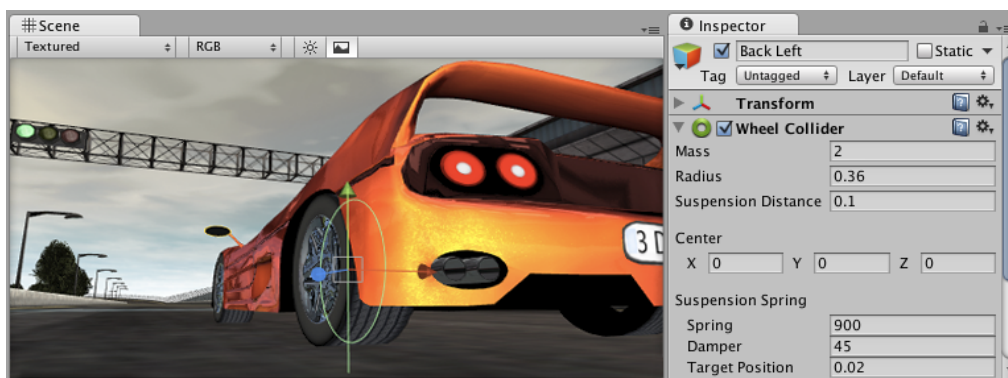
Repare que esse procedimento não gera um colisor perfeito, e sim uma aproximação. Na verdade, quando você marca a opção **Convex**, o Unity apresenta uma mensagem de erro, por meio da qual diz que o **Mesh** escolhido tem muitas regiões suavizadas e pede para você escolher um **Mesh** mais simples ou marcar a opção **Inflate**. O ideal é que o modelador 3D criador do modelo faça duas versões deste, uma para sua exibição e outra para seu **Mesh Collider**, sendo essa última com o limite máximo de 255 faces. Entretanto, o Unity faz um trabalho normalmente razoável criando esse colisor para você com a opção **Inflate**.

Wheel Collider

O **Wheel Collider** é um colisor que não é primitivo e é especialmente utilizado para rodas de veículos terrestres. Ele já vem com detecção de colisão, física para movimento das rodas, modelo de fricção das rodas, etc.

O uso do **Wheel Collider** é um assunto bastante avançado e foge do escopo mais geral deste curso. Você pode ver uma representação dele na **Figura 17**. Se você está pretendendo fazer um jogo de carros, por exemplo, precisará estudar bastante sobre o **Wheel Collider**, pois ele tem diversas propriedades para serem configuradas. Aproveite para se aprofundar mais sobre o assunto e acesse o link que está disponível na Leitura Complementar.

Figura 17 - Wheel Collider.



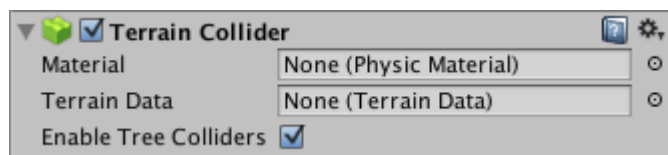
Fonte: Captura de tela do Unity – Game Engine. Disponível em:

<https://docs.unity3d.com/Manual/class-WheelCollider.html>. Acesso em: 2 de maio de 2017.

Terrain Collider

O **Terrain Collider** é um tipo de colisor que tem seu uso dedicado para o sistema de terrenos do Unity, ou seja, ele deve somente ser adicionado a terrenos e cria um colisor com o mesmo formato geométrico do terreno ao qual está relacionado. Veja na Figura 18 os atributos de um **Terrain Collider**.

Figura 18 - Atributos de um Terrain Collider.



Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://docs.unity3d.com/Manual/class-TerrainCollider.html>. Acesso em: 2 de maio de 2017.

Os atributos do Terrain Collider são:

- **Material:** o Physics Material utilizado.
- **Terrain Data:** os dados do terreno o qual ele criará no colisor.
- **Enable Tree Collider:** quando marcado, habilita colisão com árvores que estão no terreno.

O sistema de terrenos do Unity é um assunto à parte e, para obter mais informações, consulte a documentação oficial em <https://docs.unity3d.com/Manual/script-Terrain.html>

Colliders e Rigidbody

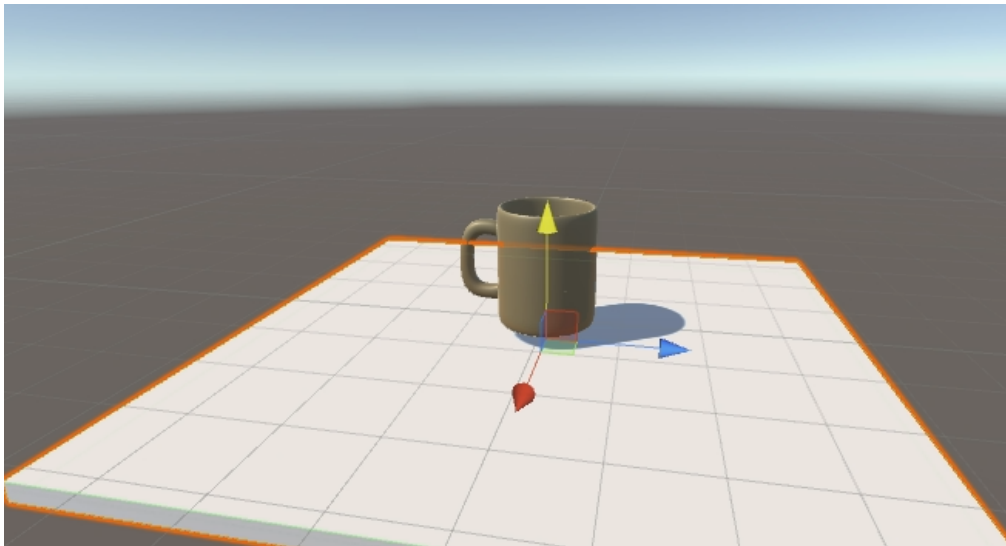
Recapitulando o que já foi visto, o componente **Rigidbody** adiciona comportamento físico a um objeto, fazendo-o ter reação a coisas, por exemplo à gravidade. Um objeto pode ter somente um **Rigidbody** (sem collider), nesse caso ele somente cairá devido à sua massa e à gravidade, passando através de qualquer outro objeto da cena.

Para um objeto ter mudança de movimento causada por colisões, é necessário ele possuir tanto um **Rigidbody** como um **Collider**. Se você deseja que somente o objeto seja um obstáculo, mas não se movimente ao receber uma colisão, basta adicionar um Collider nesse objeto.

Com a cena que estamos utilizando, e já com nossa Caneca adicionada, inicialmente removemos qualquer tipo de **Collider existente na Caneca, clicando sobre o nome do Collider no Inspector e escolhendo a opção “Remove Component”**.

Crie, agora, um Cube e posicione-o abaixo da Caneca, modificando sua escala para $X=8$, $Y=0.2$, $Z=8$, como se fosse uma mesa. Você deverá ter algo como o que está representado na **Figura 19**.

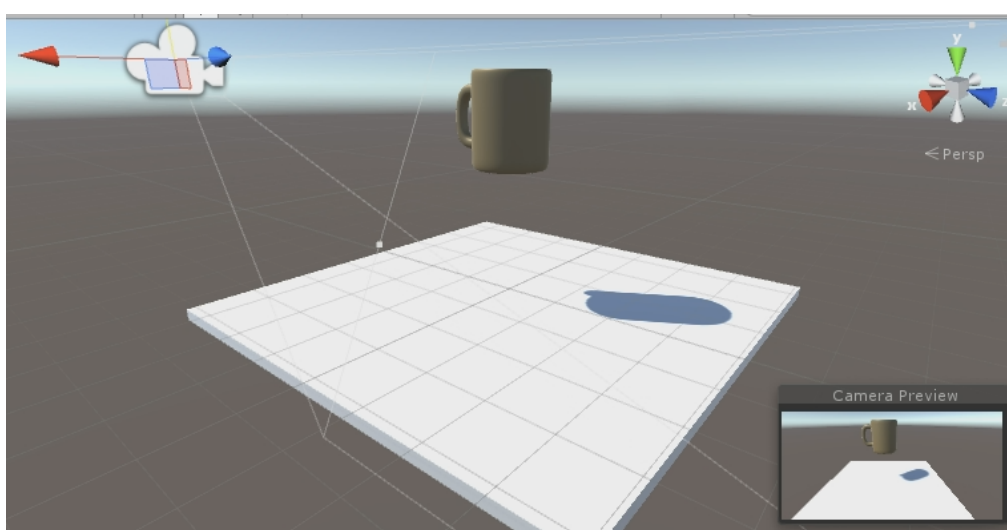
Figura 19 - Cena com uma caneca e uma mesa (um simples Cube).



Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

Posicione a Caneca um pouco acima da mesa e também aproveite para modificar a posição da sua Main Camera para que ela tenha uma visão geral de todos os elementos da cena. Você deve obter um resultado como o exemplo da **Figura 20**.

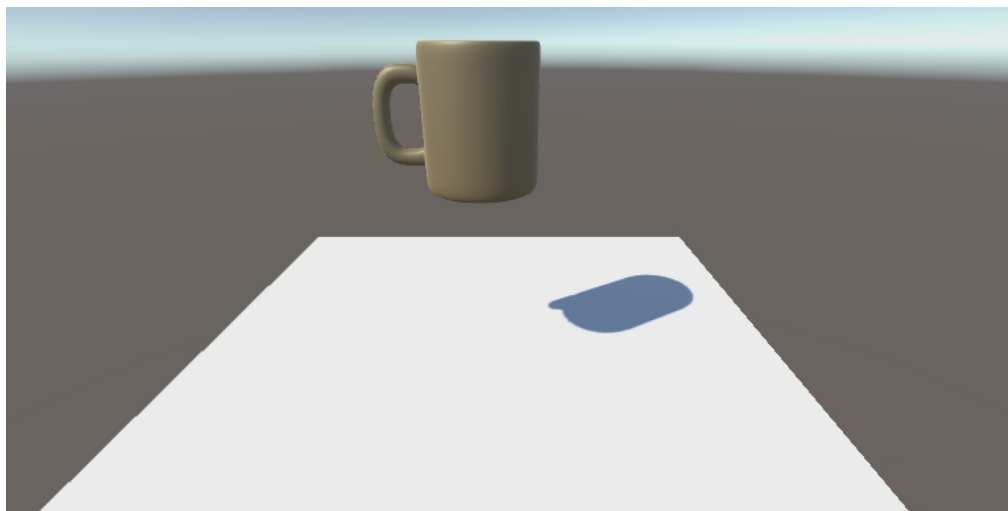
Figura 20 - Cena com câmera ajustada.



Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

Se você executar agora a cena, clicando em Play, verá que a Caneca permanece flutuando na cena, conforme mostra a **Figura 21**. Isso acontece porque os objetos não têm **Rigidbody**.

Figura 21 - Cena com objetos sem Rigidbody.

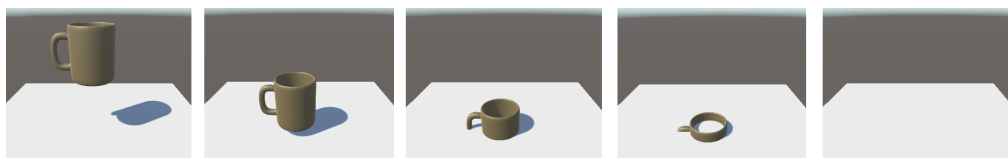


Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

Pare a execução da cena, clique na Caneca e adicione-lhe um **Rigidbody**, como vimos anteriormente. Em seguida, clique na mesa e somente se certifique que ela já tem um **Box Collider** adicionado, uma vez que o Cube utilizado como base para essa nossa mesa já é criado com um **Box Collider** por padrão.

Execute a cena agora e veja que a Caneca cairá (por ter um Rigidbody) e a mesa se manterá no mesmo local. Entretanto, algo estranho ocorrerá, pois a Caneca não parará na mesa, passando através dela, como representado na **Figura 22**.

Figura 22 - Caneca passando pela mesa.

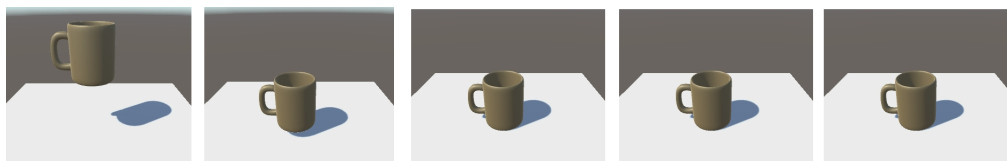


Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

Mesmo com um **Collider** na mesa, a Caneca também precisa de um collider para que aconteça uma colisão e o **Rigidbody** adicionado nela realize a reação física, fazendo-a parar sobre a mesa.

Clique na Caneca e adicione um **Box Collider**. Agora, execute novamente a cena e veja que a Caneca cai até parar sobre a mesa, como visto na **Figura 23**.

Figura 23 - Caneca com Rigidbody e Box Collider, caindo e parando sobre a mesa.



Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

Percebam que o Rigidbody adiciona a um objeto massa e reação à gravidade, enquanto o Collider informa que o objeto pode receber colisões com outros.

Assista a um vídeo que mostra como acrescentamos colisores em modelos 3D adicionados no Unity e, também, quais são os tipos de colisores disponíveis.

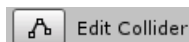


Vídeo 01 - Colliders

Colisores Combinados

Para modificar as dimensões de um Collider, basta digitar suas dimensões e posição utilizando os atributos, como Size, Center, Radius e Height, conforme vimos anteriormente. Para manipular suas propriedades utilizando o mouse na Scene

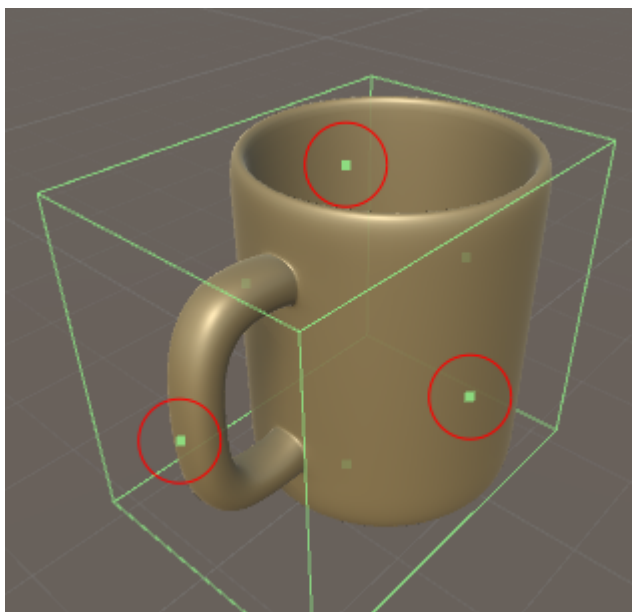
View, você pode clicar no botão “Edit Collider”



presente em cada um

deles. Pequenos quadrados verdes próximos às suas arestas aparecerão e poderão ser arrastados para modificar as suas propriedades, como visto em destaque na **Figura 24**.

Figura 24 - Detalhe mostrando os manipuladores de um Collider



Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

É possível utilizar mais de um **Collider** em um objeto, bastando adicionar no Inspector quantos você desejar. O Rigidbody utilizará todas as informações dos Colliders adicionados como se eles fossem um só colisor combinado. Isso é muito útil para criar colisores complexos utilizando somente colisores primitivos, já que você pode, por exemplo, adicionar um colisor para o corpo da Caneca e outro para o seu cabo.

Agora, veja um vídeo que mostra detalhadamente os diferentes tipos de colliders adicionados no modelo 3D utilizado na aula e os diferentes efeitos causados por eles no seu comportamento físico.

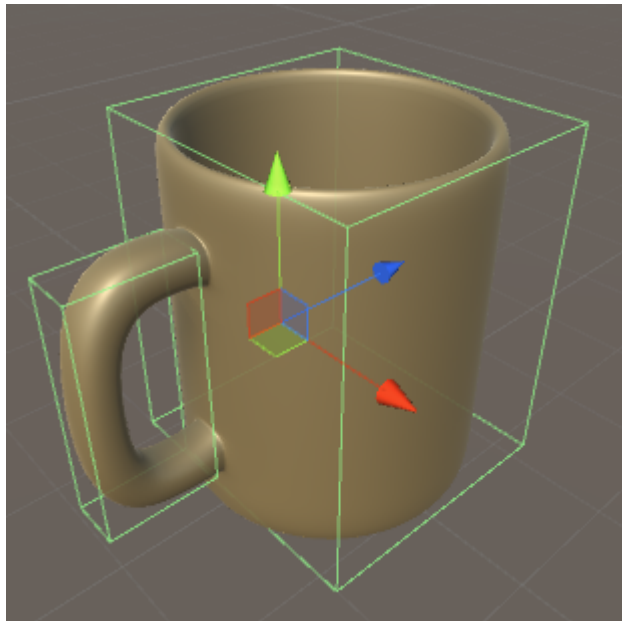


Vídeo 02 - Tipos de Colliders

Atividade 02

1. Modifique a Caneca e deixe-a exatamente com dois Box Colliders.
2. Modifique os dois Box Colliders e deixe um para o corpo da Caneca e outro para o cabo, como na imagem abaixo.

Figura 25 - Dois Box Colliders adicionados na Caneca (um para o corpo e outro para o cabo).



Fonte: Captura de tela do Unity – Game Engine. Disponível em: <https://unity3d.com/pt/>. Acesso em: 2 de maio de 2017.

Resumo

Nesta aula, aprendemos a criar Colliders em objetos 3D no Unity, assim como estudamos os tipos existentes de Colliders e suas propriedades. Aprendemos, também, mais sobre o comportamento do Rigidbody junto com os Colliders e a maneira como ocorrem as reações físicas do Unity com esses dois componentes.

Leitura Complementar

O assunto de Collider no Unity é relativamente simples, porém não tratamos, nesta aula, de alguns tipos importantes que são utilizados em determinados estilos de cenas e jogos, como o Wheel Collider e o Terrain Collider, assim como também ocorreu com o assunto de Physics Materials. Abaixo, seguem alguns links para você estudar mais esses conteúdos:

- **Wheel Collider:** <https://docs.unity3d.com/Manual/class-WheelCollider.html>
- **Terrain Collider:** <https://docs.unity3d.com/Manual/class-TerrainCollider.html>
- **Physics Material:** <https://docs.unity3d.com/Manual/class-PhysicMaterial.html>
- **3D Physics References:** <https://docs.unity3d.com/Manual/Physics3DReference.html>

Autoavaliação

1. Adicione um Cube em uma cena, remova seu colisor padrão e adicione um Sphere Collider nele, aumentando em 10% seu Radius. Logo em seguida, adicione um Rigidbody nesse objeto.
2. Adicione um outro Cube com um Rigidbody acima do primeiro Cube.

3. Execute a cena e observe o que acontece.

Referências

UNITY TECHNOLOGIES. 2016 (C). Unity 3D Online Tutorials [online]. Disponível em: <https://unity3d.com/pt/learn/tutorials> [Acessado em 16 de novembro de 2016].

UNITY TECHNOLOGIES. 2016 (C). Unity Manual - Prefabs [online]. Disponível em: <https://docs.unity3d.com/Manual/Prefabs.html> [Acessado em 16 de novembro de 2016].

STOY, C. 2006. Game object component system. In Game Programming Gems 6, Charles River Media, M. Dickheiser, Ed., Páginas 393 a 403.

MARQUES, Paulo; PEDROSO, Hernâni - C# 2.0 . Lisboa: FCA, 2005. ISBN 978-972-722 208-8

UNITY TECHNOLOGIES. 2016 (C). Unity 3D Manual [online]. Disponível em: <https://docs.unity3d.com/Manual/index.html> [Acessado em 16 de novembro de 2016].