Unity - Colisões

Collider (Colisor)

Defina a forma de um GameObject para fins de colisões físicas.

Um colisor, que é invisível, não precisa ser exatamente a mesma forma que o GameObject.

Uma aproximação da forma é muitas vezes mais eficiente e indistinguível na jogabilidade

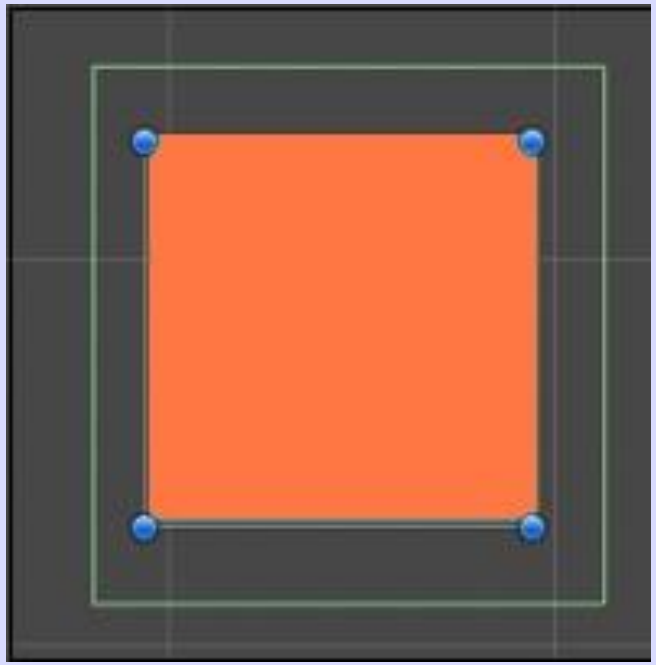
Compound Collider (Colisor Composto)

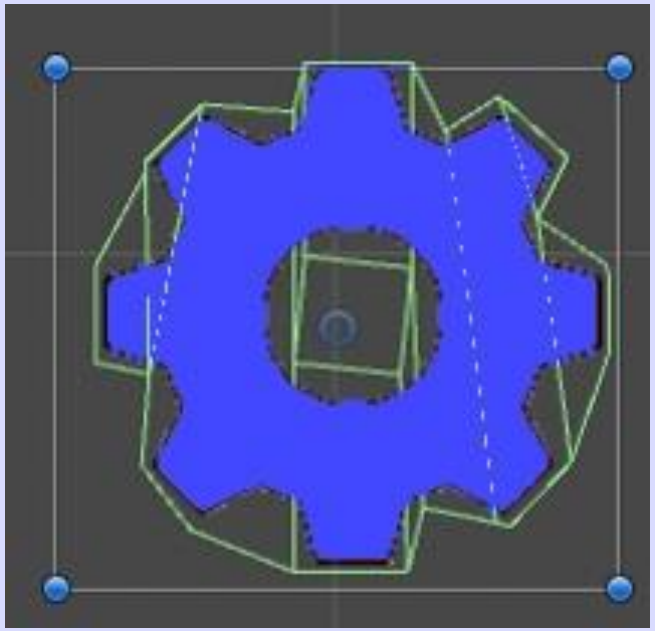
Colisores compostos aproximam a forma de um GameObject enquanto mantêm uma baixa sobrecarga no processador.

Para obter mais flexibilidade, você pode adicionar colisores adicionais em child GameObjects.

Quando você cria um colisor composto como este, você deve usar apenas um componente Rigidbody, colocado na raiz GameObject na hierarquia. Colisores primitivos não funcionam corretamente com transformações. Se você usar uma combinação de rotações e escalas não uniformes para que a forma resultante não seja mais uma forma primitiva, o colisor primitivo não pode representá-la corretamente.

Mesh Collider (Colisor de Malha)





- Em 3D, você pode usar Colisores de Malha para combinar exatamente com a forma da malha do GameObject.
- Em 2D, o Polygon Collider 2D não corresponde perfeitamente à forma do gráfico sprite, mas pode ser refinado a qualquer nível de detalhe desejado.
- Demandam muito mais processamento.
- Um colisor de malha não pode colidir com outro colisor de malha

Static colliders

Você pode adicionar colisores a um GameObject sem um componente Rigidbody para criar pisos, paredes e outros elementos imóveis de uma cena.

Estes são chamados de colisores estáticos (colisores em um GameObject que tem um Rigidbody são conhecidos como colisores dinâmicos).

Colisores estáticos podem interagir com colisores dinâmicos, mas como eles não têm um Rigidbody, eles não se movem em resposta a colisões.

Physics materials

Quando os colisores interagem, suas superfícies precisam simular as propriedades do material que deveriam representar.

Por exemplo, uma camada de gelo será escorregadia, enquanto uma bola de borracha oferecerá muito atrito e será muito saltitante.

Embora a forma dos colisores não seja deformada durante colisões, seu atrito e salto podem ser configurados usando Materiais de Física.

Acertar os parâmetros pode envolver tentativa e erro, mas um material de gelo, por exemplo, terá atrito zero (ou muito baixo) e um material de borracha com alto atrito e terá um efeito de quicar.

Triggers

O sistema de scripting pode detectar quando ocorrem colisões e iniciar ações usando a função OnCollisionEnter.

No entanto, você também pode usar o motor de física simplesmente para detectar quando um colisor entra no espaço de outro sem criar uma colisão.

Um colisor configurado como um gatilho (usando a propriedade Is Trigger) não se comporta como um objeto sólido e simplesmente permitirá que outros colisores passem.

Quando um colisor entra em seu espaço, um gatilho chamará a função OnTriggerEnter nos scripts do objeto de gatilho.

Callbacks

Na primeira atualização quando a colisão é detectada, a função OnCollisionEnter é chamada.

Durante as atualizações onde o contato é mantido, OnCollisionStay é chamado

o OnCollisionExit indica que o contato foi quebrado.

Trigger Colliders chamam as funções análogas onTriggerEnter, OnTriggerStay e OnTriggerExit

```
public class CollisionTutorialTest : MonoBehaviour
{


void OnCollisionEnter(Collision collisionInfo)
{
    print("Detected collision between " + gameObject.name + " and " +
collisionInfo.collider.name);
    print("There are " + collisionInfo.contacts.Length + " point(s) of contacts");
    print("Their relative velocity is " + collisionInfo.relativeVelocity);
}
```

```
void OnCollisionStay(Collision collisionInfo)
{
    print(gameObject.name + " and " + collisionInfo.collider.name + " are still
colliding");
}


void OnCollisionExit(Collision collisionInfo)
{
    print(gameObject.name + " and " + collisionInfo.collider.name + " are no longer
colliding");
}
}
```

```
   void OnTriggerEnter(Collider other)

{

    print("Collision detected with trigger object " + other.name);

}



void OnTriggerStay(Collider other)

{

    print("Still colliding with trigger object " + other.name);

}



void OnTriggerExit(Collider other)

{

    print(gameObject.name + " and trigger object " + other.name + " are no longer

colliding");

}
```