

Collision

Los componentes `Collider` definen la forma de un `GameObject` para propósitos de colisiones físicas. Un colisionador, que es invisible, no necesita tener exactamente la misma forma que la malla del `GameObject`. Una aproximación de la malla es a menudo más eficiente e indistinguible en el juego.

Los colisionadores más simples (y menos intensivos en procesador) son los tipos de colisionadores **primitivos**. En 3D, son el `Box Collider`, el `Sphere Collider` y el `Capsule Collider`. En 2D, puedes usar el `Box Collider 2D` y el `Circle Collider 2D`. Usted puede agregar cualquier número de estos a un solo `GameObject` para crear colisionadores compuestos.

▼ Colisionadores compuestos (Compound colliders)

Los colliders compuestos se aproximan a la forma de un `GameObject` manteniendo una baja sobrecarga del procesador. Para obtener mayor flexibilidad, puede agregar colliders adicionales en `GameObjects` hijos. Por ejemplo, usted puede rotar cajas relativas a los ejes locales del `GameObject` padre. Cuando usted crea un collider compuesto como este, usted solo debe utilizar un componente `Rigidbody`, colocado en el `GameObject` raíz en la jerarquía.

Los colisionadores primitivos no funcionan correctamente con transformaciones de corte. Si usted utiliza una combinación de rotaciones y escalas no uniformes en la jerarquía de transformación, de manera que la forma resultante ya no es una forma primitiva, el colisionador primitivo no puede representarla correctamente.

▼ Colisionadores de malla (Mesh colliders)

Hay algunos casos, sin embargo, donde incluso los colliders compuestos no son lo suficientemente precisos. En 3D, puedes usar `Mesh Colliders` para igualar la forma de la malla del `GameObject` exactamente. En 2D, el `Polygon Collider 2D` no coincide con la forma del sprite gráfico perfectamente, pero puedes refinar la forma a cualquier nivel de detalle que desees.

Estos colisionadores consumen mucho más procesador que los tipos primitivos, así que utilícelos con moderación para mantener un buen rendimiento. Además, un colisionador de malla no puede colisionar con otro colisionador de malla (es decir, no ocurre nada cuando entran en contacto). Puede evitar esto en algunos casos marcando el colisionador de malla como `Convex` en el inspector. Esto genera la forma del colisionador como un `convex hull`, que es como la malla original, pero con los recortes rellenos.

La ventaja de esto es que un colisionador de malla convexa puede colisionar con otros colisionadores de malla, por lo que puede utilizar esta función cuando tenga un personaje en movimiento con una forma adecuada. Sin embargo, una buena regla es utilizar colisionadores de malla para la geometría de la escena y aproximar la forma de `GameObjects` en movimiento utilizando colisionadores primitivos compuestos.

▼ Colisionadores estáticos (Static colliders)

Usted puede agregar colliders a un `GameObject` sin un componente `Rigidbody` para crear pisos, paredes y otros elementos inmóviles de una escena. Estos son referidos como `static colliders`. Por el contrario, los colliders en un `GameObject` que tiene un `Rigidbody` son conocidos como `dynamic colliders`. Los colliders estáticos pueden interactuar con colliders dinámicos, pero como no tienen un `Rigidbody`, no se mueven en respuesta a las colisiones.

▼ Materiales físicos (Physics materials)

Cuando los colliders interactúan, sus superficies necesitan simular las propiedades del material el cual supuestamente representan. Por ejemplo, un pedazo de hielo será resbaloso mientras una bola de caucho va a ofrecer bastante fricción y será de mucho rebote. Aunque la figura de los colliders no es deformada durante las colisiones, su fricción y rebote puede ser configurado utilizando `Physics Materials`. Obteniendo los parámetros como se debe puede involucrar un poco de intento y error, pero un material de hielo, por ejemplo, tendrá cero (o muy poca) fricción y un material de caucho tendrá mucha fricción y casi un perfecto rebote. Vea las páginas de referencia de `Physic Material` y `Physics Material 2D` para más detalles en los disponibles parámetros. Tenga en cuenta que por razones históricas, el asset 3D es llamado `Physic Material` (sin la S) pero el equivalente 2D es llamado `Physics Material 2D` (con la S).

▼ Disparadores (Triggers)

El sistema de scripting puede detectar cuando colisiones suceden e instanciar acciones utilizando la función `OnCollisionEnter`. Sin embargo, usted puede también utilizar el motor de física simplemente para detectar cuando un collider entra al espacio de otro sin crear una colisión. Un collider configurado como `Trigger` (utilizando la propiedad `Is Trigger`) no se comporta como un objeto sólido y simplemente le permitirá a otros colliders pasar a través de él. Cuando un collider entra su espacio, un trigger va a llamar la función `OnTriggerEnter` en los scripts del trigger del objeto.

▼ Llamadas de colisión para scripts (Collision callbacks for scripts)

Cuando la colisión ocurre, el motor de física llama las funciones con nombres específicos en cualquier script adjuntado a los objetos involucrados. Usted puede colocar cualquier código que usted quiera en estas funciones para responder al evento de colisión. Por ejemplo, usted podría reproducir un efecto de sonido que se estrella cuando un carro se choca a un obstáculo. En la primera actualización de física donde la colisión es detectada, la función `OnCollisionEnter` es llamada. Durante las actualizaciones donde el contacto se mantiene, `OnCollisionStay` se llama y finalmente `OnCollisionExit` indica que el contacto se ha roto. Los `Trigger colliders` llaman a las funciones análogas `OnTriggerEnter`, `OnTriggerStay` y `OnTriggerExit`. Tenga en cuenta que para físicas 2D, hay funciones equivalentes con **2D** anexado al nombre, eg, `OnCollisionEnter2D`.

Con colisiones normales, y `no-trigger`, hay un detalle adicional y es que al menos uno de los objetos involucrados debe tener un `non-kinematic Rigidbody`. Si ambos objetos son `Kinematic Rigidbodies` entonces `OnCollisionEnter`, etc, no será llamado. Con colisiones con trigger, esta restricción no aplica y ambos `kinematic` y `no-kinematic rigidbodies` van a realizar una llamada a `OnTriggerEnter` cuando entren a un trigger collider.

▼ Collider interactions

Los `Colliders` interactúan con cada uno de manera diferente, dependiendo en cómo sus `Rigidbody components` están configurados. Las tres configuraciones importantes son `Static Collider`, el `Rigidbody Collider` y el `Kinematic Rigidbody Collider`.

▼ Colisionador estático (Static Collider)

Un `static collider` es un `GameObject` que tiene un `Collider` pero no un `Rigidbody`. Los colisionadores estáticos son mayormente usados para geometría de nivel que siempre permanece en el mismo lugar y nunca se mueve. Los objetos `Rigidbody` entrantes colisionan con los colisionadores estáticos, pero no los mueven.

En casos particulares, el motor de física optimiza los colisionadores estáticos que nunca se mueven. Por ejemplo, un vehículo que descansa encima de un colisionador estático permanece dormido aunque muevas este colisionador estático. Puedes activar, desactivar o mover los colisionadores estáticos en tiempo de ejecución sin afectar especialmente a la velocidad de cálculo del motor de física. Además, puedes escalar con seguridad un `Colisionador de malla estático` siempre que la escala sea uniforme (no sesgada).

▼ Colisionador de cuerpos rígidos (Rigidbody Collider)

Este es un `GameObject` con un `Collider` y un `no-kinematic Rigidbody` normal adjunto. Los `Rigidbody colliders` son completamente simulados por el motor de física y pueden reaccionar a colisiones y fuerzas aplicadas desde un script. Estos pueden colisionar con otros objetos (incluyendo `static colliders`) y son la configuración `Collider` más comúnmente utilizada en juegos que utilizan física.

▼ Colisionador cinemático de cuerpos rígidos (Kinematic Rigidbody Collider)

Este es un `GameObject` con un `Collider` y un `kinematic Rigidbody` adjunto. Usted puede mover un objeto `kinematic ridibody` desde script modificando su componente `Transform`, pero no responderá a colisiones y fuerzas como un `no-kinematic rigidbody`. Los `Kinematic rigidbodies` deberían ser utilizados para colliders que pueden ser movidos o desactivados/activados ocasionalmente, pero que de otra manera debería comportarse como `static colliders`. Un ejemplo de esto es deslizar una puerta que normalmente debería actuar como un obstáculo físico inmovible, pero puede se puede abrir cuando sea necesario. Al contrario de un `static collider`, un `kinematic ridibody` en movimiento va a aplicar fricción a otros objetos y “despertará” otros `rigidbodies` cuando entren en contacto.

Incluso cuando `kinematic rigidbody colliders` inmovibles tienen diferente comportamiento a `static colliders`. Por ejemplo, si el `collider` es configurado como un `trigger` entonces usted también necesita agregar un `rigidbody` a él

con el fin de que reciba eventos `triggers` en su script. Si usted no quiere que el trigger caiga bajo de gravedad o de otra manera ser afectado por física, entonces usted puede configurar la propiedad `IsKinematic` en su `rigidbody`.

Un componente `Rigidbody` pueden intercambiarse entre comportamiento `normal` y `kinematic` en cualquier momento utilizando la propiedad `IsKinematic`.

Un ejemplo común de esto es el efecto `ragdoll` dónde un personaje normalmente se mueve con animaciones, pero es lanzado físicamente por una explosión o una colisión fuerte. Las extremidades del personaje pueden ser cada uno dados su propio componente `Rigidbody` con `IsKinematic` activado por defecto. Las extremidades se moverán por lo general por animaciones hasta que `IsKinematic` se desactive para todas ellas e inmediatamente se comportan como objetos de física. En este punto, una colisión o explosión va a mandar al personaje volando con sus extremidades lanzadas de una manera convincente.

▼ Matriz de acciones de colisión (Collision action matrix)

Cuando dos objetos colisionan, hay un número de diferentes eventos de script que pueden ocurrir dependiendo en las configuraciones de los `rigidbodies` de los objetos colisionando. Las tablas a continuación dan detalles de qué funciones de eventos son llamadas basándose en los componentes que están adjuntos a los objetos. Algunas combinaciones solamente causas uno de los dos objetos a ser afectados por la colisión, pero la regla general es que la física no será aplicada al objeto que no tiene un componente `Rigidbody` adjunto.

La detección de Colisión ocurre y los mensajes son mandados al momento de colisionar:

	Static Collider	Rigidbody Collider	Kinematic Rigidbody Collider	Static Trigger Collider	Rigidbody Trigger Collider	Kinematic Rigidbody Trigger Collider
Static Collider		Y				
Rigidbody Collider	Y	Y	Y			
Kinematic Rigidbody Collider		Y				
Static Trigger Collider						
Rigidbody Trigger Collider						
Kinematic Rigidbody Trigger Collider						

Los mensajes del `trigger` son mandas al momento de colisionar:

	Static Collider	Rigidbody Collider	Kinematic Rigidbody Collider	Static Trigger Collider	Rigidbody Trigger Collider	Kinematic Rigidbody Trigger Collider
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