

BLOODMOON

CAMBIO AL MOTOR FÍSICO uBODE

*Comprender las ventajas, los cambios y las soluciones
para los objetos mesh antiguos*

OpenSimulator 0.9.3.1

GUÍA PARA RESIDENTES Y PROPIETARIOS DE REGIONES

CPU

MEJORA PRINCIPAL

Reducción importante observada en
varios simuladores

RAM

MEJORA ADICIONAL

Reducción observada en varias regiones

uBODE

MOTOR PREDETERMINADO

Opción actual de la configuración de
OpenSimulator

Acerca de esta evolución

Con motivo de la actualización de BloodMoon a OpenSimulator 0.9.3.1, el motor físico BulletSim ha sido sustituido por uBODE en las regiones alojadas por la grid. Esta evolución aporta mejoras muy visibles en nuestra infraestructura, especialmente una reducción importante del uso del procesador.

PUNTO CLAVE

El cambio a uBODE no deteriora los objetos. Sin embargo, puede revelar una forma física antigua, imprecisa o demasiado compleja que BulletSim toleraba mejor anteriormente.

Índice

1	Comprender la función del motor físico
2	Por qué BloodMoon ha elegido uBODE
3	Por qué algunos meshes antiguos causan problemas
4	Reconocer los problemas más frecuentes
5	Corregir las formas físicas en Firestorm
6	Soluciones prácticas para suelos, escaleras y edificios
7	Vehículos y objetos físicos
8	Buenas prácticas y errores que deben evitarse
9	Cómo comunicar un problema
10	Conclusión: una evolución positiva para BloodMoon

1. Comprender la función del motor físico

En OpenSimulator, el motor físico gestiona todo lo relacionado con los contactos, los desplazamientos y las colisiones dentro de una región. No determina la apariencia de los objetos: calcula cómo interactúan con ellos los avatares y los demás objetos.

- impedir que un avatar atravesase una pared;
- permitir caminar sobre un suelo;
- gestionar las caídas y los obstáculos;
- desplazar los objetos físicos;
- gestionar determinados vehículos;
- calcular las colisiones entre avatares y objetos;

- permitir que determinados scripts detecten un contacto;
- gestionar escaleras, carreteras, puentes y edificios.

La apariencia y la física son dos cosas diferentes

Un objeto mesh puede contener dos representaciones distintas: el modelo visible, mostrado en el visor, y una forma física invisible utilizada por el simulador.

MODELO VISIBLE	FORMA FÍSICA
• paredes y ventanas • relieves y elementos decorativos • tiradores, tejas y molduras • detalles del mobiliario	• suelos sobre los que caminar • paredes que deben bloquear • puertas y pasos abiertos • superficies que generan una colisión

PRINCIPIO ESENCIAL

Por lo general, la forma física debería ser mucho más sencilla que el modelo visible. Una mesa muy detallada no necesita una colisión que reproduzca cada moldura.

2. Por qué BloodMoon ha elegido uBODE

uBODE es ahora el motor físico predeterminado

La configuración actual de OpenSimulator utiliza uBODE como motor físico predeterminado, junto con un sistema de cálculo de meshes diseñado para funcionar con él. Esta elección permite que BloodMoon se mantenga alineada con las evoluciones recientes del software y se prepare mejor para futuras actualizaciones.

Una reducción importante del consumo de CPU

Este es el principal beneficio observado desde el cambio. Con un contenido equivalente, varios simuladores de BloodMoon utilizan mucho menos procesador que con BulletSim.

- reducir la carga general de los servidores;
- mantener más recursos disponibles;
- limitar determinados picos de uso del procesador;
- distribuir mejor los recursos entre las regiones;
- evitar que una región muy cargada perjudique a los demás simuladores;
- facilitar el alojamiento de un gran número de regiones.

OBSERVACIÓN EN BLOODMOON

La reducción de CPU es la mejora más visible. También se ha observado una disminución de la memoria, aunque suele ser menos acusada.

Un consumo de memoria mejor controlado

Varias regiones también utilizan menos memoria desde el cambio a uBODE. Esta mejora ayuda a conservar un margen de seguridad para los eventos, los avatares y los scripts.

Una gestión más rigurosa de las colisiones

uBODE interpreta las formas físicas de manera diferente y, en ocasiones, más estricta que BulletSim. En los objetos correctamente diseñados, las colisiones pueden ser más coherentes. En algunos meshes antiguos, este mayor rigor puede revelar un defecto que ya existía en la forma física.

3. Por qué algunos meshes antiguos causan problemas

Muchos objetos mesh fueron creados hace varios años, a veces con métodos de importación diferentes de los que se utilizan actualmente. Algunos fueron probados principalmente con BulletSim.

- sin un modelo físico separado real;
- utilizando directamente el modelo visible como forma física;
- con una colisión excesivamente detallada;
- con una envoltente que rodea completamente el objeto;
- con puertas y ventanas ausentes de la física;
- con triángulos innecesarios o mal colocados;
- con una forma física desplazada respecto al modelo visible.

BulletSim podía, en ocasiones, tolerar o interpretar estos objetos de una forma que permitía utilizarlos. uBODE puede interpretarlos de otro modo y hacer aparecer colisiones invisibles, pasos cerrados o suelos desplazados.

IMPORTANTE

Un mesh antiguo no contiene una «física BulletSim». Contiene una forma física que puede interpretarse de manera diferente según el motor utilizado.

4. Reconocer los problemas más frecuentes

PROBLEMA	Una puerta o una abertura está bloqueada El objeto suele utilizar una envoltente convexa que cierra la abertura como si el edificio fuera un volumen macizo.
¿QUÉ HACER?	Probar la opción « Prim » en la parte afectada. Comprobar que solo esté seleccionada la parte enlazada correcta. Como último recurso, desactivar la colisión incorrecta y recrear las paredes con prims invisibles.
PROBLEMA	El avatar camina por encima del suelo La forma física está situada por encima del modelo visible, lo que produce la impresión de estar flotando.
¿QUÉ HACER?	Probar la opción « Prim ». Comprobar la parte correspondiente al suelo dentro de un objeto enlazado. Añadir un prim plano e invisible a la altura correcta si la colisión integrada sigue siendo incorrecta.

PROBLEMA	El avatar atraviesa el suelo La forma física puede estar ausente, desactivada o mal colocada.
¿QUÉ HACER?	Comprobar que el tipo de forma física no esté configurado como « Ninguno ». Probar la opción « Prim ». Crear una superficie invisible sencilla bajo el suelo visible.
PROBLEMA	Aparecen paredes u obstáculos invisibles La colisión puede sobresalir del modelo visible o englobar una zona que debería permanecer libre.
¿QUÉ HACER?	Seleccionar las partes enlazadas una a una. Desactivar la colisión de los elementos puramente decorativos. Sustituir la colisión compleja por algunos prims sencillos.
PROBLEMA	Las escaleras se vuelven difíciles de utilizar Cada peldaño puede interpretarse como un obstáculo independiente, especialmente si la física es muy detallada.
¿QUÉ HACER?	Conservar la escalera por su apariencia. Desactivar su colisión incorrecta si es necesario. Añadir una rampa inclinada e invisible sobre los peldaños.

PROBLEMA	Un vehículo se vuelve inestable El vehículo depende de su forma, su masa, sus scripts y el motor físico. Algunas configuraciones fueron diseñadas únicamente para BulletSim.
¿QUÉ HACER?	Rezzar una copia nueva. Reiniciar los scripts. Probar a baja velocidad en un espacio despejado. Contactar con el creador si es necesario adaptar el script.

5. Corregir las formas físicas en Firestorm

Primera precaución: conservar siempre una copia

- Compruebe que el objeto se puede copiar.
- Guarde una copia en su inventario.
- Anote su posición y su rotación.
- Realice las pruebas con una copia siempre que sea posible.
- No elimine el objeto original antes de haber validado la corrección.

Comprobar el tipo de forma física

- Haga clic con el botón derecho sobre el objeto.
- Seleccione « Editar ».
- Abra la pestaña « Características ».
- Busque la opción « Tipo de forma física ».

ENVOLVENTE CONVEXA	Forma simplificada adecuada para pequeños objetos macizos. Puede cerrar puertas, arcos, túneles o edificios huecos.
PRIM	Utiliza la forma física proporcionada durante la importación del mesh. Puede restaurar puertas y pasos si esta forma se creó correctamente.
NINGUNO	Desactiva la colisión. Es adecuado para plantas, cortinas, cuadros y elementos decorativos que los avatares pueden atravesar.

ATENCIÓN	No configure sistemáticamente todos los meshes como « Prim ». Si la forma integrada es muy compleja o está mal diseñada, puede provocar otros problemas y aumentar la carga del simulador.
-----------------	--

Modificar únicamente la parte afectada

Un edificio mesh suele estar formado por varias partes enlazadas: suelo, paredes, tejado, ventanas, escaleras y elementos decorativos. No siempre es necesario modificar el conjunto completo.

Active « Editar partes enlazadas ».

Haga clic únicamente en la parte problemática.

Compruebe su tipo de forma física.

Modifique únicamente esa parte.

Pruebe el resultado antes de continuar.

6. Soluciones prácticas para suelos, escaleras y edificios

Sustituir una física incorrecta por prims invisibles

Cuando un mesh posee una forma física inutilizable, la solución más fiable consiste en separar la apariencia visible de la colisión. El mesh permanece visible, se desactiva su colisión incorrecta y unos prims sencillos representan las superficies físicas.

- un prim aplanado para el suelo;
- algunos prims para las paredes;
- una rampa invisible para la escalera;
- espacios libres en las zonas de las puertas.

CONSEJO DE CONSTRUCCIÓN

Mantenga temporalmente visibles los prims durante su instalación. Hágalos transparentes únicamente después de comprobar su posición y los pasos.

Corregir las escaleras con una rampa invisible

Conserve la escalera mesh por su apariencia.

Desactive su colisión si provoca bloqueos.

Añada un prim inclinado sobre los peldaños.

Ajuste su inclinación y su altura.

Haga transparente el prim después de las pruebas.

El avatar subirá entonces por una superficie regular, aunque los peldaños sigan siendo visibles. Este método mejora la comodidad de desplazamiento y reduce el riesgo de bloqueos.

Corregir un suelo o una carretera

Si el avatar flota por encima del suelo:

- compruebe el tipo de forma física;
- pruebe la opción « Prim »;
- compruebe si el mesh está formado por varias partes;
- desactive la colisión incorrecta si el problema persiste;
- añada un prim plano e invisible a la altura correcta.

Si el avatar atraviesa el suelo:

- compruebe que la física no esté configurada como « Ninguno »;
- pruebe la opción « Prim »;

- compruebe que la parte seleccionada sea realmente la del suelo;
- añada una superficie invisible si la física integrada no funciona.

Aligerar los objetos decorativos

Los objetos puramente decorativos no siempre necesitan participar en los cálculos físicos. Cuando disponga de permisos de modificación, puede considerar la opción « Ninguno » para:

- plantas, hierbas y flores;
- cortinas, cuadros y decoraciones murales;
- pequeños objetos colocados sobre muebles;
- elementos fuera del alcance;
- detalles del tejado o partes puramente visuales.

IMPACTO EN EL RENDIMIENTO

Eliminar las colisiones innecesarias reduce el número de cálculos realizados por el motor físico y puede ayudar a limitar la carga de CPU de la región.

7. Vehículos y objetos físicos

Los vehículos utilizan al mismo tiempo el motor físico, su forma, su masa y sus scripts. Por ello, un vehículo diseñado para BulletSim puede necesitar una adaptación para uBODE.

- Guarde una copia del vehículo.
- Rezee una copia nueva en un espacio despejado.
- Reinicie sus scripts.
- Compruebe que únicamente las partes necesarias sean físicas.
- Compruebe que los elementos decorativos no tengan colisiones innecesarias.
- Pruebe el vehículo a baja velocidad.
- Compare su comportamiento con el observado anteriormente.

POSIBLE LIMITACIÓN

Si el vehículo sigue siendo inestable, su script puede contener parámetros ajustados específicamente para BulletSim. El equipo de BloodMoon no siempre podrá corregir un vehículo no modificable o protegido.

8. Buenas prácticas y errores que deben evitarse

Lo que no debe hacerse

- modificar un objeto sin conservar una copia;
- configurar todos los meshes de la región como « Prim » sin probarlos;
- configurar todos los objetos como « Ninguno »;
- desenlazar un edificio complejo sin guardar una copia;
- eliminar un objeto antes de haber encontrado una solución;
- hacer físicos objetos que no lo necesitan;

- utilizar cientos de pequeños prims para sustituir un suelo sencillo;
- modificar un vehículo sin conservar su configuración original;
- volver a importar un mesh sin disponer de los archivos originales o de los derechos;
- considerar automáticamente cada problema de colisión como un fallo del simulador.

¿Es necesario sustituir todos los objetos antiguos?

No. La mayoría de los objetos correctamente realizados siguen funcionando normalmente con uBODE. Solo es necesario intervenir cuando se detecta realmente un problema.

- cambiar el tipo de forma física;
- modificar únicamente una parte enlazada;
- desactivar la colisión de un elemento decorativo;
- añadir un suelo o una rampa invisible;
- sustituir una colisión incorrecta por algunos prims sencillos;
- actualizar un vehículo antiguo.

9. Cómo comunicar un problema

Si detecta un problema desde el cambio a uBODE, facilite toda la información posible al equipo de BloodMoon:

- el nombre de la región;
- las coordenadas exactas;
- el nombre del objeto;
- el nombre de su propietario;
- una descripción precisa del problema;
- una captura de pantalla, si es posible;
- una indicación de sus permisos sobre el objeto;
- la indicación de si el objeto se puede modificar o no.

Ejemplos de descripciones útiles

	EJEMPLO «La puerta principal está abierta, pero una colisión invisible impide el paso».
	EJEMPLO «El avatar camina aproximadamente cincuenta centímetros por encima de la carretera».
	EJEMPLO «El suelo es visible, pero lo atravesamos y caemos».
	EJEMPLO «Hay un obstáculo invisible delante de la escalera».

EJEMPLO

«El vehículo funcionaba antes, pero ahora vuelca inmediatamente».

Lo que el equipo podrá comprobar

- el tipo de forma física;
- la presencia de una envoltente convexa;
- una colisión incorrecta integrada en el mesh;
- una parte enlazada concreta;
- un script de vehículo;
- los permisos de modificación del objeto;
- un posible fallo real del simulador.

10. Una evolución positiva para BloodMoon

El cambio de BulletSim a uBODE representa una evolución importante para BloodMoon. Las primeras observaciones son especialmente alentadoras, con una reducción muy clara de la carga de CPU de numerosos simuladores.

También se ha observado una reducción del consumo de memoria, aunque la mejora más importante sigue siendo la del procesador. Esto permite mantener más recursos disponibles para los avatares, los scripts, los eventos y el conjunto de las regiones alojadas.

Esta transición puede requerir algunas adaptaciones en determinados meshes antiguos, pero estas dificultades afectan principalmente a objetos cuya forma física ya era imprecisa o estaba mal adaptada.

**A LARGO PLAZO, uBODE DEBE PERMITIR QUE BLOODMOON DISPONGA DE REGIONES
CON UN MENOR CONSUMO DE PROCESADOR
MEJOR ADAPTADAS A LAS VERSIONES ACTUALES DE OPENSIMULATOR
MÁS FÁCILES DE MANTENER Y MÁS ESTABLES
MEJOR PREPARADAS PARA LAS FUTURAS EVOLUCIONES DE LA GRID**

Gracias a todos los residentes, propietarios de regiones y creadores por su comprensión y por participar en la mejora de BloodMoon.