Search scripting...

unity3d.com

Manual Scripting API

Version: 2018.1 (switch to 2018.2b or 2017.4)

C# JS

RenderTargetSetup
RenderTexture
RenderTextureDescriptor
Resolution
ResourceRequest
Resources
Rigidbody
Rigidbody2D
RuntimeAnimatorController
ScalableBufferManager
Screen
ScreenCapture
ScriptableObject
Security
Shader
ShaderVariantCollection
SkeletonBone
SkinnedMeshRenderer
Skybox
SleepTimeout
SliderJoint2D
Social
SoftJointLimit
SoftJointLimitSpring
SortingLayer
SparseTexture
SphereCollider
SplatPrototype
SpringJoint
SpringJoint2D

Rigidbody

class in UnityEngine / Inherits from:Component/
Implemented in:UnityEngine.PhysicsModule

Other Versions
Leave feedback

SWITCH TO MANUAL

Description

Control of an object's position through physics simulation.

Adding a Rigidbody component to an object will put its motion under the control of Unity's physics engine. Even without adding any code, a Rigidbody object will be pulled downward by gravity and will react to collisions with incoming objects if the right Collider component is also present.

The Rigidbody also has a scripting API that lets you apply forces to the object and control it in a physically realistic way. For example, a car's behaviour can be specified in terms of the forces applied by the wheels. Given this information, the physics engine can handle most other aspects of the car's motion, so it will accelerate realistically and respond correctly to collisions.

In a script, the FixedUpdate function is recommended as the place to apply forces and change Rigidbody settings (as opposed to Update, which is used for most other frame update tasks). The reason for this is that physics updates are carried out in measured time steps that don't coincide with the frame update. FixedUpdate is called immediately before each physics update and so any changes made there will be processed directly.

A common problem when starting out with Rigidbodies is that the game physics appears to run in "slow motion". This is actually due to the scale used for your models. The default gravity settings assume that one world unit corresponds to one metre of distance. With non-physical games, it doesn't make much difference if your models are all 100 units long but when using physics, they will be treated as very large objects. If a large scale is used for objects that are supposed to be small, they will appear to fall very slowly - the physics engine thinks they are very large objects falling over very large distances. With this in mind, be sure to keep your objects more or less at their scale in real life (so a car should be about 4 units = 4 metres, for example).

Properties

angularDrag	The angular drag of the object.
angularVelocity	The angular velocity vector of the rigidbody measured in radians per second.
centerOfMass	The center of mass relative to the

constraints	Controls which degrees of freedom are allowed for the simulation of this Rigidbody.
detectCollisions	Should collision detection be enabled? (By default always enabled).
drag	The drag of the object.
freezeRotation	Controls whether physics will change the rotation of the object.
inertiaTensor	The diagonal inertia tensor of mass relative to the center of mass.
inertiaTensorRotation	The rotation of the inertia tensor.
interpolation	Interpolation allows you to smooth out the effect of running physics at a fixed frame rate.
isKinematic	Controls whether physics affects the rigidbody.
mass	The mass of the rigidbody.
maxAngularVelocity	The maximimum angular velocity of the rigidbody. (Default 7) range { 0, infinity }.
maxDepenetrationVelocity	Maximum velocity of a rigidbody when moving out of penetrating state.
position	The position of the rigidbody.
rotation	The rotation of the rigidbody.
sleepThreshold	The mass-normalized energy threshold, below which objects start going to sleep.
solverIterations	The solverIterations determines how accurately Rigidbody joints and collision contacts are resolved. Overrides Physics.defaultSolverIterations. Must be positive.
solverVelocityIterations	The solverVelocityIterations affects how how accurately Rigidbody joints and collision contacts are resolved. Overrides Physics.defaultSolverVelocityIterations. Must be positive.
useGravity	Controls whether gravity affects this rigidbody.
velocity	The velocity vector of the rigidbody.
worldCenterOfMass	The center of mass of the rigidbody in world space (Read Only).

AddForce	Adds a force to the Rigidbody.
AddForceAtPosition	Applies force at position. As a result this will apply a torque and force on the object.
AddRelativeForce	Adds a force to the rigidbody relative to its coordinate system.
AddRelativeTorque	Adds a torque to the rigidbody relative to its coordinate system.
AddTorque	Adds a torque to the rigidbody.
ClosestPointOnBounds	The closest point to the bounding box of the attached colliders.
GetPointVelocity	The velocity of the rigidbody at the point worldPoint in global space.
GetRelativePointVelocity	The velocity relative to the rigidbody at the point relativePoint.
IsSleeping	Is the rigidbody sleeping?
MovePosition	Moves the rigidbody to position.
MoveRotation	Rotates the rigidbody to rotation.
ResetCenterOfMass	Reset the center of mass of the rigidbody.
ResetInertiaTensor	Reset the inertia tensor value and rotation.
SetDensity	Sets the mass based on the attached colliders assuming a constant density.
Sleep	Forces a rigidbody to sleep at least one frame.
SweepTest	Tests if a rigidbody would collide with anything, if it was moved through the scene.
SweepTestAll	Like Rigidbody.SweepTest, but returns all hits.
WakeUp	Forces a rigidbody to wake up.

Messages

OnCollisionEnter	OnCollisionEnter is called when this collider/rigidbody has begun touching another rigidbody/collider.
OnCollisionExit	OnCollisionEnter is called when this collider/rigidbody has stopped touching another rigidbody/collider.

Inherited Members

Properties

gameObject	The game object this component is attached to. A component is always attached to a game object.
tag	The tag of this game object.
transform	The Transform attached to this GameObject.
hideFlags	Should the object be hidden, saved with the scene or modifiable by the user?
name	The name of the object.

Public Methods

BroadcastMessage	Calls the method named methodName on every MonoBehaviour in this game object or any of its children.
CompareTag	Is this game object tagged with tag ?
GetComponent	Returns the component of Type type if the game object has one attached, null if it doesn't.
GetComponentInChildren	Returns the component of Type type in the GameObject or any of its children using depth first search.
GetComponentInParent	Returns the component of Type type in the GameObject or any of its parents.
GetComponents	Returns all components of Type type in the GameObject.
GetComponentsInChildren	Returns all components of Type type in the GameObject or any of its children.
GetComponentsInParent	Returns all components of Type type in the GameObject or any of its parents.
SendMessage	Calls the method named methodName on every MonoBehaviour in this game object.
SendMessageUpwards	Calls the method named methodName on every MonoBehaviour in this game

ToString	Returns the name of the GameObject.

Static Methods

Destroy	Removes a gameobject, component or asset.
DestroyImmediate	Destroys the object obj immediately. You are strongly recommended to use Destroy instead.
DontDestroyOnLoad	Makes the object target not be destroyed automatically when loading a new scene.
FindObjectOfType	Returns the first active loaded object of Type type.
FindObjectsOfType	Returns a list of all active loaded objects of Type type.
Instantiate	Clones the object original and returns the clone.

Operators

bool	Does the object exist?
operator !=	Compares if two objects refer to a different object.
operator ==	Compares two object references to see if they refer to the same object.

Did you find this page useful? Please give it a rating:

Report a problem on this page

Is something described here not working as you expect it to? It might be a **Known Issue**. Please check with the Issue Tracker at **issuetracker.unity3d.com**.

Tutorials Community Answers Knowledge Base Forums

Asset Store