Api documentation

**Collider2D**

class in UnityEngine

/

Inherits from:[Behaviour](https://docs.unity3d.com/ScriptReference/Behaviour.html)

/

Implemented in:[UnityEngine.Physics2DModule](https://docs.unity3d.com/ScriptReference/UnityEngine.Physics2DModule.html)

**Description**

Parent class for collider types used with 2D gameplay.

See Also: [BoxCollider2D](https://docs.unity3d.com/ScriptReference/BoxCollider2D.html), [CircleCollider2D](https://docs.unity3d.com/ScriptReference/CircleCollider2D.html), [PolygonCollider2D](https://docs.unity3d.com/ScriptReference/PolygonCollider2D.html), [EdgeCollider2D](https://docs.unity3d.com/ScriptReference/EdgeCollider2D.html).

**Properties**

|  |  |
| --- | --- |
| [attachedRigidbody](https://docs.unity3d.com/ScriptReference/Collider2D-attachedRigidbody.html) | The Rigidbody2D attached to the Collider2D. |
| [bounciness](https://docs.unity3d.com/ScriptReference/Collider2D-bounciness.html) | Get the bounciness used by the collider. |
| [bounds](https://docs.unity3d.com/ScriptReference/Collider2D-bounds.html) | The world space bounding area of the collider. |
| [composite](https://docs.unity3d.com/ScriptReference/Collider2D-composite.html) | Get the CompositeCollider2D that is available to be attached to the collider. |
| [density](https://docs.unity3d.com/ScriptReference/Collider2D-density.html) | The density of the collider used to calculate its mass (when auto mass is enabled). |
| [friction](https://docs.unity3d.com/ScriptReference/Collider2D-friction.html) | Get the friction used by the collider. |
| [isTrigger](https://docs.unity3d.com/ScriptReference/Collider2D-isTrigger.html) | Is this collider configured as a trigger? |
| [offset](https://docs.unity3d.com/ScriptReference/Collider2D-offset.html) | The local offset of the collider geometry. |
| [shapeCount](https://docs.unity3d.com/ScriptReference/Collider2D-shapeCount.html) | The number of separate shaped regions in the collider. |
| [sharedMaterial](https://docs.unity3d.com/ScriptReference/Collider2D-sharedMaterial.html) | The PhysicsMaterial2D that is applied to this collider. |
| [usedByComposite](https://docs.unity3d.com/ScriptReference/Collider2D-usedByComposite.html) | Sets whether the Collider will be used or not used by a CompositeCollider2D. |
| [usedByEffector](https://docs.unity3d.com/ScriptReference/Collider2D-usedByEffector.html) | Whether the collider is used by an attached effector or not. |

**Public Methods**

|  |  |
| --- | --- |
| [Cast](https://docs.unity3d.com/ScriptReference/Collider2D.Cast.html) | Casts the Collider shape into the Scene starting at the Collider position ignoring the Collider itself. |
| [ClosestPoint](https://docs.unity3d.com/ScriptReference/Collider2D.ClosestPoint.html) | Returns a point on the perimeter of this Collider that is closest to the specified position. |
| [CreateMesh](https://docs.unity3d.com/ScriptReference/Collider2D.CreateMesh.html) | Creates a planar Mesh that is identical to the area defined by the Collider2D geometry. |
| [Distance](https://docs.unity3d.com/ScriptReference/Collider2D.Distance.html) | Calculates the minimum separation of this collider against another collider. |
| [GetContacts](https://docs.unity3d.com/ScriptReference/Collider2D.GetContacts.html) | Retrieves all contact points for this Collider. |
| [GetShapeHash](https://docs.unity3d.com/ScriptReference/Collider2D.GetShapeHash.html) | Generates a simple hash value based upon the geometry of the Collider2D. |
| [IsTouching](https://docs.unity3d.com/ScriptReference/Collider2D.IsTouching.html) | Check whether this collider is touching the collider or not. |
| [IsTouchingLayers](https://docs.unity3d.com/ScriptReference/Collider2D.IsTouchingLayers.html) | Checks whether this collider is touching any colliders on the specified layerMask or not. |
| [OverlapCollider](https://docs.unity3d.com/ScriptReference/Collider2D.OverlapCollider.html) | Get a list of all colliders that overlap this collider. |
| [OverlapPoint](https://docs.unity3d.com/ScriptReference/Collider2D.OverlapPoint.html) | Check if a collider overlaps a point in space. |
| [Raycast](https://docs.unity3d.com/ScriptReference/Collider2D.Raycast.html) | Casts a ray into the Scene that starts at the Collider position and ignores the Collider itself. |

**Messages**

|  |  |
| --- | --- |
| [OnCollisionEnter2D](https://docs.unity3d.com/ScriptReference/Collider2D.OnCollisionEnter2D.html) | Sent when an incoming collider makes contact with this object's collider (2D physics only). |
| [OnCollisionExit2D](https://docs.unity3d.com/ScriptReference/Collider2D.OnCollisionExit2D.html) | Sent when a collider on another object stops touching this object's collider (2D physics only). |
| [OnCollisionStay2D](https://docs.unity3d.com/ScriptReference/Collider2D.OnCollisionStay2D.html) | Sent each frame where a collider on another object is touching this object's collider (2D physics only). |
| [OnTriggerEnter2D](https://docs.unity3d.com/ScriptReference/Collider2D.OnTriggerEnter2D.html) | Sent when another object enters a trigger collider attached to this object (2D physics only). |
| [OnTriggerExit2D](https://docs.unity3d.com/ScriptReference/Collider2D.OnTriggerExit2D.html) | Sent when another object leaves a trigger collider attached to this object (2D physics only). |
| [OnTriggerStay2D](https://docs.unity3d.com/ScriptReference/Collider2D.OnTriggerStay2D.html) | Sent each frame where another object is within a trigger collider attached to this object (2D physics only). |

**Inherited Members**

**Properties**

|  |  |
| --- | --- |
| [enabled](https://docs.unity3d.com/ScriptReference/Behaviour-enabled.html) | Enabled Behaviours are Updated, disabled Behaviours are not. |
| [isActiveAndEnabled](https://docs.unity3d.com/ScriptReference/Behaviour-isActiveAndEnabled.html) | Has the Behaviour had active and enabled called? |
| [gameObject](https://docs.unity3d.com/ScriptReference/Component-gameObject.html) | The game object this component is attached to. A component is always attached to a game object. |
| [tag](https://docs.unity3d.com/ScriptReference/Component-tag.html) | The tag of this game object. |
| [transform](https://docs.unity3d.com/ScriptReference/Component-transform.html) | The Transform attached to this GameObject. |
| [hideFlags](https://docs.unity3d.com/ScriptReference/Object-hideFlags.html) | Should the object be hidden, saved with the Scene or modifiable by the user? |
| [name](https://docs.unity3d.com/ScriptReference/Object-name.html) | The name of the object. |

**Public Methods**

|  |  |
| --- | --- |
| [BroadcastMessage](https://docs.unity3d.com/ScriptReference/Component.BroadcastMessage.html) | Calls the method named methodName on every MonoBehaviour in this game object or any of its children. |
| [CompareTag](https://docs.unity3d.com/ScriptReference/Component.CompareTag.html) | Is this game object tagged with tag ? |
| [GetComponent](https://docs.unity3d.com/ScriptReference/Component.GetComponent.html) | Returns the component of Type type if the game object has one attached, null if it doesn't. |
| [GetComponentInChildren](https://docs.unity3d.com/ScriptReference/Component.GetComponentInChildren.html) | Returns the component of Type type in the GameObject or any of its children using depth first search. |
| [GetComponentInParent](https://docs.unity3d.com/ScriptReference/Component.GetComponentInParent.html) | Returns the component of Type type in the GameObject or any of its parents. |
| [GetComponents](https://docs.unity3d.com/ScriptReference/Component.GetComponents.html) | Returns all components of Type type in the GameObject. |
| [GetComponentsInChildren](https://docs.unity3d.com/ScriptReference/Component.GetComponentsInChildren.html) | Returns all components of Type type in the GameObject or any of its children. |
| [GetComponentsInParent](https://docs.unity3d.com/ScriptReference/Component.GetComponentsInParent.html) | Returns all components of Type type in the GameObject or any of its parents. |
| [SendMessage](https://docs.unity3d.com/ScriptReference/Component.SendMessage.html) | Calls the method named methodName on every MonoBehaviour in this game object. |
| [SendMessageUpwards](https://docs.unity3d.com/ScriptReference/Component.SendMessageUpwards.html) | Calls the method named methodName on every MonoBehaviour in this game object and on every ancestor of the behaviour. |
| [TryGetComponent](https://docs.unity3d.com/ScriptReference/Component.TryGetComponent.html) | Gets the component of the specified type, if it exists. |
| [GetInstanceID](https://docs.unity3d.com/ScriptReference/Object.GetInstanceID.html) | Returns the instance id of the object. |
| [ToString](https://docs.unity3d.com/ScriptReference/Object.ToString.html) | Returns the name of the object. |

**Static Methods**

|  |  |
| --- | --- |
| [Destroy](https://docs.unity3d.com/ScriptReference/Object.Destroy.html) | Removes a GameObject, component or asset. |
| [DestroyImmediate](https://docs.unity3d.com/ScriptReference/Object.DestroyImmediate.html) | Destroys the object obj immediately. You are strongly recommended to use Destroy instead. |
| [DontDestroyOnLoad](https://docs.unity3d.com/ScriptReference/Object.DontDestroyOnLoad.html) | Do not destroy the target Object when loading a new Scene. |
| [FindObjectOfType](https://docs.unity3d.com/ScriptReference/Object.FindObjectOfType.html) | Returns the first active loaded object of Type type. |
| [FindObjectsOfType](https://docs.unity3d.com/ScriptReference/Object.FindObjectsOfType.html) | Returns a list of all active loaded objects of Type type. |
| [Instantiate](https://docs.unity3d.com/ScriptReference/Object.Instantiate.html) | Clones the object original and returns the clone. |

**Operators**

|  |  |
| --- | --- |
| [bool](https://docs.unity3d.com/ScriptReference/Object-operator_Object.html) | Does the object exist? |
| [operator !=](https://docs.unity3d.com/ScriptReference/Object-operator_ne.html) | Compares if two objects refer to a different object. |
| [operator ==](https://docs.unity3d.com/ScriptReference/Object-operator_eq.html) | Compares two object references to see if they refer to the same object. |

**Transform**

class in UnityEngine

/

Inherits from:[Component](https://docs.unity3d.com/ScriptReference/Component.html)

/

Implemented in:[UnityEngine.CoreModule](https://docs.unity3d.com/ScriptReference/UnityEngine.CoreModule.html)

**Description**

Position, rotation and scale of an object.

Every object in a Scene has a Transform. It's used to store and manipulate the position, rotation and scale of the object. Every Transform can have a parent, which allows you to apply position, rotation and scale hierarchically. This is the hierarchy seen in the Hierarchy pane. They also support enumerators so you can loop through children using:

using UnityEngine;  
  
public class Example : [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)

{

// Moves all transform children 10 units upwards!

void Start()

{

foreach ([Transform](https://docs.unity3d.com/ScriptReference/Transform.html) child in transform)

{

child.position += [Vector3.up](https://docs.unity3d.com/ScriptReference/Vector3-up.html) \* 10.0f;

}

}

}

See Also: [The component reference](https://docs.unity3d.com/Manual/class-Transform.html), [Physics](https://docs.unity3d.com/ScriptReference/Physics.html) class.

**Properties**

|  |  |
| --- | --- |
| [childCount](https://docs.unity3d.com/ScriptReference/Transform-childCount.html) | The number of children the parent Transform has. |
| [eulerAngles](https://docs.unity3d.com/ScriptReference/Transform-eulerAngles.html) | The rotation as Euler angles in degrees. |
| [forward](https://docs.unity3d.com/ScriptReference/Transform-forward.html) | Returns a normalized vector representing the blue axis of the transform in world space. |
| [hasChanged](https://docs.unity3d.com/ScriptReference/Transform-hasChanged.html) | Has the transform changed since the last time the flag was set to 'false'? |
| [hierarchyCapacity](https://docs.unity3d.com/ScriptReference/Transform-hierarchyCapacity.html) | The transform capacity of the transform's hierarchy data structure. |
| [hierarchyCount](https://docs.unity3d.com/ScriptReference/Transform-hierarchyCount.html) | The number of transforms in the transform's hierarchy data structure. |
| [localEulerAngles](https://docs.unity3d.com/ScriptReference/Transform-localEulerAngles.html) | The rotation as Euler angles in degrees relative to the parent transform's rotation. |
| [localPosition](https://docs.unity3d.com/ScriptReference/Transform-localPosition.html) | Position of the transform relative to the parent transform. |
| [localRotation](https://docs.unity3d.com/ScriptReference/Transform-localRotation.html) | The rotation of the transform relative to the transform rotation of the parent. |
| [localScale](https://docs.unity3d.com/ScriptReference/Transform-localScale.html) | The scale of the transform relative to the GameObjects parent. |
| [localToWorldMatrix](https://docs.unity3d.com/ScriptReference/Transform-localToWorldMatrix.html) | Matrix that transforms a point from local space into world space (Read Only). |
| [lossyScale](https://docs.unity3d.com/ScriptReference/Transform-lossyScale.html) | The global scale of the object (Read Only). |
| [parent](https://docs.unity3d.com/ScriptReference/Transform-parent.html) | The parent of the transform. |
| [position](https://docs.unity3d.com/ScriptReference/Transform-position.html) | The world space position of the Transform. |
| [right](https://docs.unity3d.com/ScriptReference/Transform-right.html) | The red axis of the transform in world space. |
| [root](https://docs.unity3d.com/ScriptReference/Transform-root.html) | Returns the topmost transform in the hierarchy. |
| [rotation](https://docs.unity3d.com/ScriptReference/Transform-rotation.html) | A Quaternion that stores the rotation of the Transform in world space. |
| [up](https://docs.unity3d.com/ScriptReference/Transform-up.html) | The green axis of the transform in world space. |
| [worldToLocalMatrix](https://docs.unity3d.com/ScriptReference/Transform-worldToLocalMatrix.html) | Matrix that transforms a point from world space into local space (Read Only). |

**Public Methods**

|  |  |
| --- | --- |
| [DetachChildren](https://docs.unity3d.com/ScriptReference/Transform.DetachChildren.html) | Unparents all children. |
| [Find](https://docs.unity3d.com/ScriptReference/Transform.Find.html) | Finds a child by n and returns it. |
| [GetChild](https://docs.unity3d.com/ScriptReference/Transform.GetChild.html) | Returns a transform child by index. |
| [GetSiblingIndex](https://docs.unity3d.com/ScriptReference/Transform.GetSiblingIndex.html) | Gets the sibling index. |
| [InverseTransformDirection](https://docs.unity3d.com/ScriptReference/Transform.InverseTransformDirection.html) | Transforms a direction from world space to local space. The opposite of Transform.TransformDirection. |
| [InverseTransformPoint](https://docs.unity3d.com/ScriptReference/Transform.InverseTransformPoint.html) | Transforms position from world space to local space. |
| [InverseTransformVector](https://docs.unity3d.com/ScriptReference/Transform.InverseTransformVector.html) | Transforms a vector from world space to local space. The opposite of Transform.TransformVector. |
| [IsChildOf](https://docs.unity3d.com/ScriptReference/Transform.IsChildOf.html) | Is this transform a child of parent? |
| [LookAt](https://docs.unity3d.com/ScriptReference/Transform.LookAt.html) | Rotates the transform so the forward vector points at /target/'s current position. |
| [Rotate](https://docs.unity3d.com/ScriptReference/Transform.Rotate.html) | Use Transform.Rotate to rotate GameObjects in a variety of ways. The rotation is often provided as an Euler angle and not a Quaternion. |
| [RotateAround](https://docs.unity3d.com/ScriptReference/Transform.RotateAround.html) | Rotates the transform about axis passing through point in world coordinates by angle degrees. |
| [SetAsFirstSibling](https://docs.unity3d.com/ScriptReference/Transform.SetAsFirstSibling.html) | Move the transform to the start of the local transform list. |
| [SetAsLastSibling](https://docs.unity3d.com/ScriptReference/Transform.SetAsLastSibling.html) | Move the transform to the end of the local transform list. |
| [SetParent](https://docs.unity3d.com/ScriptReference/Transform.SetParent.html) | Set the parent of the transform. |
| [SetPositionAndRotation](https://docs.unity3d.com/ScriptReference/Transform.SetPositionAndRotation.html) | Sets the world space position and rotation of the Transform component. |
| [SetSiblingIndex](https://docs.unity3d.com/ScriptReference/Transform.SetSiblingIndex.html) | Sets the sibling index. |
| [TransformDirection](https://docs.unity3d.com/ScriptReference/Transform.TransformDirection.html) | Transforms direction from local space to world space. |
| [TransformPoint](https://docs.unity3d.com/ScriptReference/Transform.TransformPoint.html) | Transforms position from local space to world space. |
| [TransformVector](https://docs.unity3d.com/ScriptReference/Transform.TransformVector.html) | Transforms vector from local space to world space. |
| [Translate](https://docs.unity3d.com/ScriptReference/Transform.Translate.html) | Moves the transform in the direction and distance of translation. |

**Inherited Members**

**Properties**

|  |  |
| --- | --- |
| [gameObject](https://docs.unity3d.com/ScriptReference/Component-gameObject.html) | The game object this component is attached to. A component is always attached to a game object. |
| [tag](https://docs.unity3d.com/ScriptReference/Component-tag.html) | The tag of this game object. |
| [transform](https://docs.unity3d.com/ScriptReference/Component-transform.html) | The Transform attached to this GameObject. |
| [hideFlags](https://docs.unity3d.com/ScriptReference/Object-hideFlags.html) | Should the object be hidden, saved with the Scene or modifiable by the user? |
| [name](https://docs.unity3d.com/ScriptReference/Object-name.html) | The name of the object. |

**Public Methods**

|  |  |
| --- | --- |
| [BroadcastMessage](https://docs.unity3d.com/ScriptReference/Component.BroadcastMessage.html) | Calls the method named methodName on every MonoBehaviour in this game object or any of its children. |
| [CompareTag](https://docs.unity3d.com/ScriptReference/Component.CompareTag.html) | Is this game object tagged with tag ? |
| [GetComponent](https://docs.unity3d.com/ScriptReference/Component.GetComponent.html) | Returns the component of Type type if the game object has one attached, null if it doesn't. |
| [GetComponentInChildren](https://docs.unity3d.com/ScriptReference/Component.GetComponentInChildren.html) | Returns the component of Type type in the GameObject or any of its children using depth first search. |
| [GetComponentInParent](https://docs.unity3d.com/ScriptReference/Component.GetComponentInParent.html) | Returns the component of Type type in the GameObject or any of its parents. |
| [GetComponents](https://docs.unity3d.com/ScriptReference/Component.GetComponents.html) | Returns all components of Type type in the GameObject. |
| [GetComponentsInChildren](https://docs.unity3d.com/ScriptReference/Component.GetComponentsInChildren.html) | Returns all components of Type type in the GameObject or any of its children. |
| [GetComponentsInParent](https://docs.unity3d.com/ScriptReference/Component.GetComponentsInParent.html) | Returns all components of Type type in the GameObject or any of its parents. |
| [SendMessage](https://docs.unity3d.com/ScriptReference/Component.SendMessage.html) | Calls the method named methodName on every MonoBehaviour in this game object. |
| [SendMessageUpwards](https://docs.unity3d.com/ScriptReference/Component.SendMessageUpwards.html) | Calls the method named methodName on every MonoBehaviour in this game object and on every ancestor of the behaviour. |
| [TryGetComponent](https://docs.unity3d.com/ScriptReference/Component.TryGetComponent.html) | Gets the component of the specified type, if it exists. |
| [GetInstanceID](https://docs.unity3d.com/ScriptReference/Object.GetInstanceID.html) | Returns the instance id of the object. |
| [ToString](https://docs.unity3d.com/ScriptReference/Object.ToString.html) | Returns the name of the object. |

**Static Methods**

|  |  |
| --- | --- |
| [Destroy](https://docs.unity3d.com/ScriptReference/Object.Destroy.html) | Removes a GameObject, component or asset. |
| [DestroyImmediate](https://docs.unity3d.com/ScriptReference/Object.DestroyImmediate.html) | Destroys the object obj immediately. You are strongly recommended to use Destroy instead. |
| [DontDestroyOnLoad](https://docs.unity3d.com/ScriptReference/Object.DontDestroyOnLoad.html) | Do not destroy the target Object when loading a new Scene. |
| [FindObjectOfType](https://docs.unity3d.com/ScriptReference/Object.FindObjectOfType.html) | Returns the first active loaded object of Type type. |
| [FindObjectsOfType](https://docs.unity3d.com/ScriptReference/Object.FindObjectsOfType.html) | Returns a list of all active loaded objects of Type type. |
| [Instantiate](https://docs.unity3d.com/ScriptReference/Object.Instantiate.html) | Clones the object original and returns the clone. |

**Operators**

|  |  |
| --- | --- |
| [bool](https://docs.unity3d.com/ScriptReference/Object-operator_Object.html) | Does the object exist? |
| [operator !=](https://docs.unity3d.com/ScriptReference/Object-operator_ne.html) | Compares if two objects refer to a different object. |
| [operator ==](https://docs.unity3d.com/ScriptReference/Object-operator_eq.html) | Compares two object references to see if they refer to the same object. |

**Vector2**

struct in UnityEngine

/

Implemented in:[UnityEngine.CoreModule](https://docs.unity3d.com/ScriptReference/UnityEngine.CoreModule.html)

**Description**

Representation of 2D vectors and points.

This structure is used in some places to represent 2D positions and vectors (e.g. texture coordinates in a [Mesh](https://docs.unity3d.com/ScriptReference/Mesh.html) or texture offsets in [Material](https://docs.unity3d.com/ScriptReference/Material.html)). In the majority of other cases a Vector3 is used.

**Static Properties**

|  |  |
| --- | --- |
| [down](https://docs.unity3d.com/ScriptReference/Vector2-down.html) | Shorthand for writing Vector2(0, -1). |
| [left](https://docs.unity3d.com/ScriptReference/Vector2-left.html) | Shorthand for writing Vector2(-1, 0). |
| [negativeInfinity](https://docs.unity3d.com/ScriptReference/Vector2-negativeInfinity.html) | Shorthand for writing Vector2(float.NegativeInfinity, float.NegativeInfinity). |
| [one](https://docs.unity3d.com/ScriptReference/Vector2-one.html) | Shorthand for writing Vector2(1, 1). |
| [positiveInfinity](https://docs.unity3d.com/ScriptReference/Vector2-positiveInfinity.html) | Shorthand for writing Vector2(float.PositiveInfinity, float.PositiveInfinity). |
| [right](https://docs.unity3d.com/ScriptReference/Vector2-right.html) | Shorthand for writing Vector2(1, 0). |
| [up](https://docs.unity3d.com/ScriptReference/Vector2-up.html) | Shorthand for writing Vector2(0, 1). |
| [zero](https://docs.unity3d.com/ScriptReference/Vector2-zero.html) | Shorthand for writing Vector2(0, 0). |

**Properties**

|  |  |
| --- | --- |
| [magnitude](https://docs.unity3d.com/ScriptReference/Vector2-magnitude.html) | Returns the length of this vector (Read Only). |
| [normalized](https://docs.unity3d.com/ScriptReference/Vector2-normalized.html) | Returns this vector with a magnitude of 1 (Read Only). |
| [sqrMagnitude](https://docs.unity3d.com/ScriptReference/Vector2-sqrMagnitude.html) | Returns the squared length of this vector (Read Only). |
| [this[int]](https://docs.unity3d.com/ScriptReference/Vector2.Index_operator.html) | Access the x or y component using [0] or [1] respectively. |
| [x](https://docs.unity3d.com/ScriptReference/Vector2-x.html) | X component of the vector. |
| [y](https://docs.unity3d.com/ScriptReference/Vector2-y.html) | Y component of the vector. |

**Constructors**

|  |  |
| --- | --- |
| [Vector2](https://docs.unity3d.com/ScriptReference/Vector2-ctor.html) | Constructs a new vector with given x, y components. |

**Public Methods**

|  |  |
| --- | --- |
| [Equals](https://docs.unity3d.com/ScriptReference/Vector2.Equals.html) | Returns true if the given vector is exactly equal to this vector. |
| [Normalize](https://docs.unity3d.com/ScriptReference/Vector2.Normalize.html) | Makes this vector have a magnitude of 1. |
| [Set](https://docs.unity3d.com/ScriptReference/Vector2.Set.html) | Set x and y components of an existing Vector2. |
| [ToString](https://docs.unity3d.com/ScriptReference/Vector2.ToString.html) | Returns a nicely formatted string for this vector. |

**Static Methods**

|  |  |
| --- | --- |
| [Angle](https://docs.unity3d.com/ScriptReference/Vector2.Angle.html) | Returns the unsigned angle in degrees between from and to. |
| [ClampMagnitude](https://docs.unity3d.com/ScriptReference/Vector2.ClampMagnitude.html) | Returns a copy of vector with its magnitude clamped to maxLength. |
| [Distance](https://docs.unity3d.com/ScriptReference/Vector2.Distance.html) | Returns the distance between a and b. |
| [Dot](https://docs.unity3d.com/ScriptReference/Vector2.Dot.html) | Dot Product of two vectors. |
| [Lerp](https://docs.unity3d.com/ScriptReference/Vector2.Lerp.html) | Linearly interpolates between vectors a and b by t. |
| [LerpUnclamped](https://docs.unity3d.com/ScriptReference/Vector2.LerpUnclamped.html) | Linearly interpolates between vectors a and b by t. |
| [Max](https://docs.unity3d.com/ScriptReference/Vector2.Max.html) | Returns a vector that is made from the largest components of two vectors. |
| [Min](https://docs.unity3d.com/ScriptReference/Vector2.Min.html) | Returns a vector that is made from the smallest components of two vectors. |
| [MoveTowards](https://docs.unity3d.com/ScriptReference/Vector2.MoveTowards.html) | Moves a point current towards target. |
| [Perpendicular](https://docs.unity3d.com/ScriptReference/Vector2.Perpendicular.html) | Returns the 2D vector perpendicular to this 2D vector. The result is always rotated 90-degrees in a counter-clockwise direction for a 2D coordinate system where the positive Y axis goes up. |
| [Reflect](https://docs.unity3d.com/ScriptReference/Vector2.Reflect.html) | Reflects a vector off the vector defined by a normal. |
| [Scale](https://docs.unity3d.com/ScriptReference/Vector2.Scale.html) | Multiplies two vectors component-wise. |
| [SignedAngle](https://docs.unity3d.com/ScriptReference/Vector2.SignedAngle.html) | Returns the signed angle in degrees between from and to. |
| [SmoothDamp](https://docs.unity3d.com/ScriptReference/Vector2.SmoothDamp.html) | Gradually changes a vector towards a desired goal over time. |

**Operators**

|  |  |
| --- | --- |
| [operator -](https://docs.unity3d.com/ScriptReference/Vector2-operator_subtract.html) | Subtracts one vector from another. |
| [operator \*](https://docs.unity3d.com/ScriptReference/Vector2-operator_multiply.html) | Multiplies a vector by a number. |
| [operator /](https://docs.unity3d.com/ScriptReference/Vector2-operator_divide.html) | Divides a vector by a number. |
| [operator +](https://docs.unity3d.com/ScriptReference/Vector2-operator_add.html) | Adds two vectors. |
| [operator ==](https://docs.unity3d.com/ScriptReference/Vector2-operator_eq.html) | Returns true if two vectors are approximately equal. |
| [Vector2](https://docs.unity3d.com/ScriptReference/Vector2-operator_Vector3.html) | Converts a Vector3 to a Vector2. |
| [Vector3](https://docs.unity3d.com/ScriptReference/Vector2-operator_Vector2.html) | Converts a Vector2 to a Vector3. |

**Rigidbody2D**

class in UnityEngine

/

Inherits from:[Component](https://docs.unity3d.com/ScriptReference/Component.html)

/

Implemented in:[UnityEngine.Physics2DModule](https://docs.unity3d.com/ScriptReference/UnityEngine.Physics2DModule.html)

**Description**

Rigidbody physics component for 2D sprites.

The Rigidbody2D class essentially provides the same functionality in 2D that the Rigidbody class provides in 3D. Adding a Rigidbody2D component to a sprite puts it under the control of the physics engine. By itself, this means that the sprite will be affected by gravity and can be controlled from scripts using forces. By adding the appropriate collider component, the sprite will also respond to collisions with other sprites. This behaviour comes entirely from Unity's physics system; very little code is required to get impressive and authentic physical behaviour and allows for "emergent" gameplay that was not explicitly coded into the game.  
  
See Also: [Rigidbody](https://docs.unity3d.com/ScriptReference/Rigidbody.html) class, [SpriteRenderer](https://docs.unity3d.com/ScriptReference/SpriteRenderer.html) class, [Collider2D](https://docs.unity3d.com/ScriptReference/Collider2D.html) class, [Joint2D](https://docs.unity3d.com/ScriptReference/Joint2D.html) class.

**Properties**

|  |  |
| --- | --- |
| [angularDrag](https://docs.unity3d.com/ScriptReference/Rigidbody2D-angularDrag.html) | Coefficient of angular drag. |
| [angularVelocity](https://docs.unity3d.com/ScriptReference/Rigidbody2D-angularVelocity.html) | Angular velocity in degrees per second. |
| [attachedColliderCount](https://docs.unity3d.com/ScriptReference/Rigidbody2D-attachedColliderCount.html) | Returns the number of Collider2D attached to this Rigidbody2D. |
| [bodyType](https://docs.unity3d.com/ScriptReference/Rigidbody2D-bodyType.html) | The physical behaviour type of the Rigidbody2D. |
| [centerOfMass](https://docs.unity3d.com/ScriptReference/Rigidbody2D-centerOfMass.html) | The center of mass of the rigidBody in local space. |
| [collisionDetectionMode](https://docs.unity3d.com/ScriptReference/Rigidbody2D-collisionDetectionMode.html) | The method used by the physics engine to check if two objects have collided. |
| [constraints](https://docs.unity3d.com/ScriptReference/Rigidbody2D-constraints.html) | Controls which degrees of freedom are allowed for the simulation of this Rigidbody2D. |
| [drag](https://docs.unity3d.com/ScriptReference/Rigidbody2D-drag.html) | Coefficient of drag. |
| [freezeRotation](https://docs.unity3d.com/ScriptReference/Rigidbody2D-freezeRotation.html) | Controls whether physics will change the rotation of the object. |
| [gravityScale](https://docs.unity3d.com/ScriptReference/Rigidbody2D-gravityScale.html) | The degree to which this object is affected by gravity. |
| [inertia](https://docs.unity3d.com/ScriptReference/Rigidbody2D-inertia.html) | The rigidBody rotational inertia. |
| [interpolation](https://docs.unity3d.com/ScriptReference/Rigidbody2D-interpolation.html) | Physics interpolation used between updates. |
| [isKinematic](https://docs.unity3d.com/ScriptReference/Rigidbody2D-isKinematic.html) | Should this rigidbody be taken out of physics control? |
| [mass](https://docs.unity3d.com/ScriptReference/Rigidbody2D-mass.html) | Mass of the Rigidbody. |
| [position](https://docs.unity3d.com/ScriptReference/Rigidbody2D-position.html) | The position of the rigidbody. |
| [rotation](https://docs.unity3d.com/ScriptReference/Rigidbody2D-rotation.html) | The rotation of the rigidbody. |
| [sharedMaterial](https://docs.unity3d.com/ScriptReference/Rigidbody2D-sharedMaterial.html) | The PhysicsMaterial2D that is applied to all Collider2D attached to this Rigidbody2D. |
| [simulated](https://docs.unity3d.com/ScriptReference/Rigidbody2D-simulated.html) | Indicates whether the rigid body should be simulated or not by the physics system. |
| [sleepMode](https://docs.unity3d.com/ScriptReference/Rigidbody2D-sleepMode.html) | The sleep state that the rigidbody will initially be in. |
| [useAutoMass](https://docs.unity3d.com/ScriptReference/Rigidbody2D-useAutoMass.html) | Should the total rigid-body mass be automatically calculated from the [[Collider2D.density]] of attached colliders? |
| [useFullKinematicContacts](https://docs.unity3d.com/ScriptReference/Rigidbody2D-useFullKinematicContacts.html) | Should kinematic/kinematic and kinematic/static collisions be allowed? |
| [velocity](https://docs.unity3d.com/ScriptReference/Rigidbody2D-velocity.html) | Linear velocity of the Rigidbody in units per second. |
| [worldCenterOfMass](https://docs.unity3d.com/ScriptReference/Rigidbody2D-worldCenterOfMass.html) | Gets the center of mass of the rigidBody in global space. |

**Public Methods**

|  |  |
| --- | --- |
| [AddForce](https://docs.unity3d.com/ScriptReference/Rigidbody2D.AddForce.html) | Apply a force to the rigidbody. |
| [AddForceAtPosition](https://docs.unity3d.com/ScriptReference/Rigidbody2D.AddForceAtPosition.html) | Apply a force at a given position in space. |
| [AddRelativeForce](https://docs.unity3d.com/ScriptReference/Rigidbody2D.AddRelativeForce.html) | Adds a force to the rigidbody2D relative to its coordinate system. |
| [AddTorque](https://docs.unity3d.com/ScriptReference/Rigidbody2D.AddTorque.html) | Apply a torque at the rigidbody's centre of mass. |
| [Cast](https://docs.unity3d.com/ScriptReference/Rigidbody2D.Cast.html) | All the Collider2D shapes attached to the Rigidbody2D are cast into the Scene starting at each Collider position ignoring the Colliders attached to the same Rigidbody2D. |
| [ClosestPoint](https://docs.unity3d.com/ScriptReference/Rigidbody2D.ClosestPoint.html) | Returns a point on the perimeter of all enabled Colliders attached to this Rigidbody that is closest to the specified position. |
| [Distance](https://docs.unity3d.com/ScriptReference/Rigidbody2D.Distance.html) | Calculates the minimum distance of this collider against all Collider2D attached to this Rigidbody2D. |
| [GetAttachedColliders](https://docs.unity3d.com/ScriptReference/Rigidbody2D.GetAttachedColliders.html) | Returns all Collider2D that are attached to this Rigidbody2D. |
| [GetContacts](https://docs.unity3d.com/ScriptReference/Rigidbody2D.GetContacts.html) | Retrieves all contact points for all of the Collider(s) attached to this Rigidbody. |
| [GetPoint](https://docs.unity3d.com/ScriptReference/Rigidbody2D.GetPoint.html) | Get a local space point given the point point in rigidBody global space. |
| [GetPointVelocity](https://docs.unity3d.com/ScriptReference/Rigidbody2D.GetPointVelocity.html) | The velocity of the rigidbody at the point Point in global space. |
| [GetRelativePoint](https://docs.unity3d.com/ScriptReference/Rigidbody2D.GetRelativePoint.html) | Get a global space point given the point relativePoint in rigidBody local space. |
| [GetRelativePointVelocity](https://docs.unity3d.com/ScriptReference/Rigidbody2D.GetRelativePointVelocity.html) | The velocity of the rigidbody at the point Point in local space. |
| [GetRelativeVector](https://docs.unity3d.com/ScriptReference/Rigidbody2D.GetRelativeVector.html) | Get a global space vector given the vector relativeVector in rigidBody local space. |
| [GetVector](https://docs.unity3d.com/ScriptReference/Rigidbody2D.GetVector.html) | Get a local space vector given the vector vector in rigidBody global space. |
| [IsAwake](https://docs.unity3d.com/ScriptReference/Rigidbody2D.IsAwake.html) | Is the rigidbody "awake"? |
| [IsSleeping](https://docs.unity3d.com/ScriptReference/Rigidbody2D.IsSleeping.html) | Is the rigidbody "sleeping"? |
| [IsTouching](https://docs.unity3d.com/ScriptReference/Rigidbody2D.IsTouching.html) | Checks whether the collider is touching any of the collider(s) attached to this rigidbody or not. |
| [IsTouchingLayers](https://docs.unity3d.com/ScriptReference/Rigidbody2D.IsTouchingLayers.html) | Checks whether any of the collider(s) attached to this rigidbody are touching any colliders on the specified layerMask or not. |
| [MovePosition](https://docs.unity3d.com/ScriptReference/Rigidbody2D.MovePosition.html) | Moves the rigidbody to position. |
| [MoveRotation](https://docs.unity3d.com/ScriptReference/Rigidbody2D.MoveRotation.html) | Rotates the Rigidbody to angle (given in degrees). |
| [OverlapCollider](https://docs.unity3d.com/ScriptReference/Rigidbody2D.OverlapCollider.html) | Get a list of all Colliders that overlap all Colliders attached to this Rigidbody2D. |
| [OverlapPoint](https://docs.unity3d.com/ScriptReference/Rigidbody2D.OverlapPoint.html) | Check if any of the Rigidbody2D colliders overlap a point in space. |
| [SetRotation](https://docs.unity3d.com/ScriptReference/Rigidbody2D.SetRotation.html) | Sets the rotation of the Rigidbody2D to angle (given in degrees). |
| [Sleep](https://docs.unity3d.com/ScriptReference/Rigidbody2D.Sleep.html) | Make the rigidbody "sleep". |
| [WakeUp](https://docs.unity3d.com/ScriptReference/Rigidbody2D.WakeUp.html) | Disables the "sleeping" state of a rigidbody. |

**Inherited Members**

**Properties**

|  |  |
| --- | --- |
| [gameObject](https://docs.unity3d.com/ScriptReference/Component-gameObject.html) | The game object this component is attached to. A component is always attached to a game object. |
| [tag](https://docs.unity3d.com/ScriptReference/Component-tag.html) | The tag of this game object. |
| [transform](https://docs.unity3d.com/ScriptReference/Component-transform.html) | The Transform attached to this GameObject. |
| [hideFlags](https://docs.unity3d.com/ScriptReference/Object-hideFlags.html) | Should the object be hidden, saved with the Scene or modifiable by the user? |
| [name](https://docs.unity3d.com/ScriptReference/Object-name.html) | The name of the object. |

**Public Methods**

|  |  |
| --- | --- |
| [BroadcastMessage](https://docs.unity3d.com/ScriptReference/Component.BroadcastMessage.html) | Calls the method named methodName on every MonoBehaviour in this game object or any of its children. |
| [CompareTag](https://docs.unity3d.com/ScriptReference/Component.CompareTag.html) | Is this game object tagged with tag ? |
| [GetComponent](https://docs.unity3d.com/ScriptReference/Component.GetComponent.html) | Returns the component of Type type if the game object has one attached, null if it doesn't. |
| [GetComponentInChildren](https://docs.unity3d.com/ScriptReference/Component.GetComponentInChildren.html) | Returns the component of Type type in the GameObject or any of its children using depth first search. |
| [GetComponentInParent](https://docs.unity3d.com/ScriptReference/Component.GetComponentInParent.html) | Returns the component of Type type in the GameObject or any of its parents. |
| [GetComponents](https://docs.unity3d.com/ScriptReference/Component.GetComponents.html) | Returns all components of Type type in the GameObject. |
| [GetComponentsInChildren](https://docs.unity3d.com/ScriptReference/Component.GetComponentsInChildren.html) | Returns all components of Type type in the GameObject or any of its children. |
| [GetComponentsInParent](https://docs.unity3d.com/ScriptReference/Component.GetComponentsInParent.html) | Returns all components of Type type in the GameObject or any of its parents. |
| [SendMessage](https://docs.unity3d.com/ScriptReference/Component.SendMessage.html) | Calls the method named methodName on every MonoBehaviour in this game object. |
| [SendMessageUpwards](https://docs.unity3d.com/ScriptReference/Component.SendMessageUpwards.html) | Calls the method named methodName on every MonoBehaviour in this game object and on every ancestor of the behaviour. |
| [TryGetComponent](https://docs.unity3d.com/ScriptReference/Component.TryGetComponent.html) | Gets the component of the specified type, if it exists. |
| [GetInstanceID](https://docs.unity3d.com/ScriptReference/Object.GetInstanceID.html) | Returns the instance id of the object. |
| [ToString](https://docs.unity3d.com/ScriptReference/Object.ToString.html) | Returns the name of the object. |

**Static Methods**

|  |  |
| --- | --- |
| [Destroy](https://docs.unity3d.com/ScriptReference/Object.Destroy.html) | Removes a GameObject, component or asset. |
| [DestroyImmediate](https://docs.unity3d.com/ScriptReference/Object.DestroyImmediate.html) | Destroys the object obj immediately. You are strongly recommended to use Destroy instead. |
| [DontDestroyOnLoad](https://docs.unity3d.com/ScriptReference/Object.DontDestroyOnLoad.html) | Do not destroy the target Object when loading a new Scene. |
| [FindObjectOfType](https://docs.unity3d.com/ScriptReference/Object.FindObjectOfType.html) | Returns the first active loaded object of Type type. |
| [FindObjectsOfType](https://docs.unity3d.com/ScriptReference/Object.FindObjectsOfType.html) | Returns a list of all active loaded objects of Type type. |
| [Instantiate](https://docs.unity3d.com/ScriptReference/Object.Instantiate.html) | Clones the object original and returns the clone. |

**Operators**

|  |  |
| --- | --- |
| [bool](https://docs.unity3d.com/ScriptReference/Object-operator_Object.html) | Does the object exist? |
| [operator !=](https://docs.unity3d.com/ScriptReference/Object-operator_ne.html) | Compares if two objects refer to a different object. |
| [operator ==](https://docs.unity3d.com/ScriptReference/Object-operator_eq.html) | Compares two object references to see if they refer to the same object. |

**MonoBehaviour**

class in UnityEngine

/

Inherits from:[Behaviour](https://docs.unity3d.com/ScriptReference/Behaviour.html)

/

Implemented in:[UnityEngine.CoreModule](https://docs.unity3d.com/ScriptReference/UnityEngine.CoreModule.html)

**Description**

MonoBehaviour is the base class from which every Unity script derives.

When you use C#, you must explicitly derive from MonoBehaviour.  
  
This class doesn't support the [null-conditional operator](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/expressions#null-conditional-operator)(**?.**) and the [null-coalescing operator](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/expressions#the-null-coalescing-operator)(**??**).  
  
For code samples, see the individual MonoBehaviour methods.  
  
**Note:** There is a checkbox for enabling or disabling MonoBehaviour in the Unity Editor. It disables functions when unticked. If none of these functions are present in the script, the Unity Editor does not display the checkbox. The functions are:  
  
[Start](https://docs.unity3d.com/ScriptReference/MonoBehaviour.Start.html)()  
[Update](https://docs.unity3d.com/ScriptReference/MonoBehaviour.Update.html)()  
[FixedUpdate](https://docs.unity3d.com/ScriptReference/MonoBehaviour.FixedUpdate.html)()  
[LateUpdate](https://docs.unity3d.com/ScriptReference/MonoBehaviour.LateUpdate.html)()  
[OnGUI](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnGUI.html)()  
[OnDisable](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnDisable.html)()  
[OnEnable](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnEnable.html)()  
  
See Also: The [Deactivating GameObjects](https://docs.unity3d.com/Manual/DeactivatingGameObjects.html) page in the manual.

**Properties**

|  |  |
| --- | --- |
| [runInEditMode](https://docs.unity3d.com/ScriptReference/MonoBehaviour-runInEditMode.html) | Allow a specific instance of a MonoBehaviour to run in edit mode (only available in the editor). |
| [useGUILayout](https://docs.unity3d.com/ScriptReference/MonoBehaviour-useGUILayout.html) | Disabling this lets you skip the GUI layout phase. |

**Public Methods**

|  |  |
| --- | --- |
| [CancelInvoke](https://docs.unity3d.com/ScriptReference/MonoBehaviour.CancelInvoke.html) | Cancels all Invoke calls on this MonoBehaviour. |
| [Invoke](https://docs.unity3d.com/ScriptReference/MonoBehaviour.Invoke.html) | Invokes the method methodName in time seconds. |
| [InvokeRepeating](https://docs.unity3d.com/ScriptReference/MonoBehaviour.InvokeRepeating.html) | Invokes the method methodName in time seconds, then repeatedly every repeatRate seconds. |
| [IsInvoking](https://docs.unity3d.com/ScriptReference/MonoBehaviour.IsInvoking.html) | Is any invoke on methodName pending? |
| [StartCoroutine](https://docs.unity3d.com/ScriptReference/MonoBehaviour.StartCoroutine.html) | Starts a Coroutine. |
| [StopAllCoroutines](https://docs.unity3d.com/ScriptReference/MonoBehaviour.StopAllCoroutines.html) | Stops all coroutines running on this behaviour. |
| [StopCoroutine](https://docs.unity3d.com/ScriptReference/MonoBehaviour.StopCoroutine.html) | Stops the first coroutine named methodName, or the coroutine stored in routine running on this behaviour. |

**Static Methods**

|  |  |
| --- | --- |
| [print](https://docs.unity3d.com/ScriptReference/MonoBehaviour-print.html) | Logs message to the Unity Console (identical to Debug.Log). |

**Messages**

|  |  |
| --- | --- |
| [Awake](https://docs.unity3d.com/ScriptReference/MonoBehaviour.Awake.html) | Awake is called when the script instance is being loaded. |
| [FixedUpdate](https://docs.unity3d.com/ScriptReference/MonoBehaviour.FixedUpdate.html) | Frame-rate independent MonoBehaviour.FixedUpdate message for physics calculations. |
| [LateUpdate](https://docs.unity3d.com/ScriptReference/MonoBehaviour.LateUpdate.html) | LateUpdate is called every frame, if the Behaviour is enabled. |
| [OnAnimatorIK](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnAnimatorIK.html) | Callback for setting up animation IK (inverse kinematics). |
| [OnAnimatorMove](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnAnimatorMove.html) | Callback for processing animation movements for modifying root motion. |
| [OnApplicationFocus](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnApplicationFocus.html) | Sent to all GameObjects when the player gets or loses focus. |
| [OnApplicationPause](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnApplicationPause.html) | Sent to all GameObjects when the application pauses. |
| [OnApplicationQuit](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnApplicationQuit.html) | Sent to all GameObjects before the application quits. |
| [OnAudioFilterRead](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnAudioFilterRead.html) | If OnAudioFilterRead is implemented, Unity will insert a custom filter into the audio DSP chain. |
| [OnBecameInvisible](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnBecameInvisible.html) | OnBecameInvisible is called when the renderer is no longer visible by any camera. |
| [OnBecameVisible](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnBecameVisible.html) | OnBecameVisible is called when the renderer became visible by any camera. |
| [OnCollisionEnter](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnCollisionEnter.html) | OnCollisionEnter is called when this collider/rigidbody has begun touching another rigidbody/collider. |
| [OnCollisionEnter2D](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnCollisionEnter2D.html) | Sent when an incoming collider makes contact with this object's collider (2D physics only). |
| [OnCollisionExit](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnCollisionExit.html) | OnCollisionExit is called when this collider/rigidbody has stopped touching another rigidbody/collider. |
| [OnCollisionExit2D](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnCollisionExit2D.html) | Sent when a collider on another object stops touching this object's collider (2D physics only). |
| [OnCollisionStay](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnCollisionStay.html) | :ref::OnCollisionStay is called once per frame for every collider/rigidbody that is touching rigidbody/collider. |
| [OnCollisionStay2D](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnCollisionStay2D.html) | Sent each frame where a collider on another object is touching this object's collider (2D physics only). |
| [OnConnectedToServer](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnConnectedToServer.html) | Called on the client when you have successfully connected to a server. |
| [OnControllerColliderHit](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnControllerColliderHit.html) | OnControllerColliderHit is called when the controller hits a collider while performing a Move. |
| [OnDestroy](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnDestroy.html) | Destroying the attached Behaviour will result in the game or Scene receiving OnDestroy. |
| [OnDisable](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnDisable.html) | This function is called when the behaviour becomes disabled. |
| [OnDisconnectedFromServer](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnDisconnectedFromServer.html) | Called on the client when the connection was lost or you disconnected from the server. |
| [OnDrawGizmos](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnDrawGizmos.html) | Implement OnDrawGizmos if you want to draw gizmos that are also pickable and always drawn. |
| [OnDrawGizmosSelected](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnDrawGizmosSelected.html) | Implement OnDrawGizmosSelected to draw a gizmo if the object is selected. |
| [OnEnable](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnEnable.html) | This function is called when the object becomes enabled and active. |
| [OnFailedToConnect](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnFailedToConnect.html) | Called on the client when a connection attempt fails for some reason. |
| [OnFailedToConnectToMasterServer](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnFailedToConnectToMasterServer.html) | Called on clients or servers when there is a problem connecting to the MasterServer. |
| [OnGUI](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnGUI.html) | OnGUI is called for rendering and handling GUI events. |
| [OnJointBreak](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnJointBreak.html) | Called when a joint attached to the same game object broke. |
| [OnJointBreak2D](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnJointBreak2D.html) | Called when a Joint2D attached to the same game object breaks. |
| [OnMasterServerEvent](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnMasterServerEvent.html) | Called on clients or servers when reporting events from the MasterServer. |
| [OnMouseDown](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnMouseDown.html) | OnMouseDown is called when the user has pressed the mouse button while over the Collider. |
| [OnMouseDrag](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnMouseDrag.html) | OnMouseDrag is called when the user has clicked on a Collider and is still holding down the mouse. |
| [OnMouseEnter](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnMouseEnter.html) | Called when the mouse enters the Collider. |
| [OnMouseExit](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnMouseExit.html) | Called when the mouse is not any longer over the Collider. |
| [OnMouseOver](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnMouseOver.html) | Called every frame while the mouse is over the Collider. |
| [OnMouseUp](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnMouseUp.html) | OnMouseUp is called when the user has released the mouse button. |
| [OnMouseUpAsButton](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnMouseUpAsButton.html) | OnMouseUpAsButton is only called when the mouse is released over the same Collider as it was pressed. |
| [OnNetworkInstantiate](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnNetworkInstantiate.html) | Called on objects which have been network instantiated with Network.Instantiate. |
| [OnParticleCollision](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnParticleCollision.html) | OnParticleCollision is called when a particle hits a Collider. |
| [OnParticleSystemStopped](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnParticleSystemStopped.html) | OnParticleSystemStopped is called when all particles in the system have died, and no new particles will be born. New particles cease to be created either after Stop is called, or when the duration property of a non-looping system has been exceeded. |
| [OnParticleTrigger](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnParticleTrigger.html) | OnParticleTrigger is called when any particles in a Particle System meet the conditions in the trigger module. |
| [OnParticleUpdateJobScheduled](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnParticleUpdateJobScheduled.html) | OnParticleUpdateJobScheduled is called when a Particle System's built-in update job has been scheduled. |
| [OnPlayerConnected](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnPlayerConnected.html) | Called on the server whenever a new player has successfully connected. |
| [OnPlayerDisconnected](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnPlayerDisconnected.html) | Called on the server whenever a player disconnected from the server. |
| [OnPostRender](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnPostRender.html) | OnPostRender is called after a camera finished rendering the Scene. |
| [OnPreCull](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnPreCull.html) | OnPreCull is called before a camera culls the Scene. |
| [OnPreRender](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnPreRender.html) | OnPreRender is called before a camera starts rendering the Scene. |
| [OnRenderImage](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnRenderImage.html) | OnRenderImage is called after all rendering is complete to render image. |
| [OnRenderObject](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnRenderObject.html) | OnRenderObject is called after camera has rendered the Scene. |
| [OnSerializeNetworkView](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnSerializeNetworkView.html) | Used to customize synchronization of variables in a script watched by a network view. |
| [OnServerInitialized](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnServerInitialized.html) | Called on the server whenever a Network.InitializeServer was invoked and has completed. |
| [OnTransformChildrenChanged](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnTransformChildrenChanged.html) | This function is called when the list of children of the transform of the GameObject has changed. |
| [OnTransformParentChanged](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnTransformParentChanged.html) | This function is called when the parent property of the transform of the GameObject has changed. |
| [OnTriggerEnter](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnTriggerEnter.html) | When a GameObject collides with another GameObject, Unity calls OnTriggerEnter. |
| [OnTriggerEnter2D](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnTriggerEnter2D.html) | Sent when another object enters a trigger collider attached to this object (2D physics only). |
| [OnTriggerExit](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnTriggerExit.html) | OnTriggerExit is called when the Collider other has stopped touching the trigger. |
| [OnTriggerExit2D](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnTriggerExit2D.html) | Sent when another object leaves a trigger collider attached to this object (2D physics only). |
| [OnTriggerStay](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnTriggerStay.html) | OnTriggerStay is called once per physics update for every Collider other that is touching the trigger. |
| [OnTriggerStay2D](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnTriggerStay2D.html) | Sent each frame where another object is within a trigger collider attached to this object (2D physics only). |
| [OnValidate](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnValidate.html) | This function is called when the script is loaded or a value is changed in the Inspector (Called in the editor only). |
| [OnWillRenderObject](https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnWillRenderObject.html) | OnWillRenderObject is called for each camera if the object is visible and not a UI element. |
| [Reset](https://docs.unity3d.com/ScriptReference/MonoBehaviour.Reset.html) | Reset to default values. |
| [Start](https://docs.unity3d.com/ScriptReference/MonoBehaviour.Start.html) | Start is called on the frame when a script is enabled just before any of the Update methods are called the first time. |
| [Update](https://docs.unity3d.com/ScriptReference/MonoBehaviour.Update.html) | Update is called every frame, if the MonoBehaviour is enabled. |

**Inherited Members**

**Properties**

|  |  |
| --- | --- |
| [enabled](https://docs.unity3d.com/ScriptReference/Behaviour-enabled.html) | Enabled Behaviours are Updated, disabled Behaviours are not. |
| [isActiveAndEnabled](https://docs.unity3d.com/ScriptReference/Behaviour-isActiveAndEnabled.html) | Has the Behaviour had active and enabled called? |
| [gameObject](https://docs.unity3d.com/ScriptReference/Component-gameObject.html) | The game object this component is attached to. A component is always attached to a game object. |
| [tag](https://docs.unity3d.com/ScriptReference/Component-tag.html) | The tag of this game object. |
| [transform](https://docs.unity3d.com/ScriptReference/Component-transform.html) | The Transform attached to this GameObject. |
| [hideFlags](https://docs.unity3d.com/ScriptReference/Object-hideFlags.html) | Should the object be hidden, saved with the Scene or modifiable by the user? |
| [name](https://docs.unity3d.com/ScriptReference/Object-name.html) | The name of the object. |

**Public Methods**

|  |  |
| --- | --- |
| [BroadcastMessage](https://docs.unity3d.com/ScriptReference/Component.BroadcastMessage.html) | Calls the method named methodName on every MonoBehaviour in this game object or any of its children. |
| [CompareTag](https://docs.unity3d.com/ScriptReference/Component.CompareTag.html) | Is this game object tagged with tag ? |
| [GetComponent](https://docs.unity3d.com/ScriptReference/Component.GetComponent.html) | Returns the component of Type type if the game object has one attached, null if it doesn't. |
| [GetComponentInChildren](https://docs.unity3d.com/ScriptReference/Component.GetComponentInChildren.html) | Returns the component of Type type in the GameObject or any of its children using depth first search. |
| [GetComponentInParent](https://docs.unity3d.com/ScriptReference/Component.GetComponentInParent.html) | Returns the component of Type type in the GameObject or any of its parents. |
| [GetComponents](https://docs.unity3d.com/ScriptReference/Component.GetComponents.html) | Returns all components of Type type in the GameObject. |
| [GetComponentsInChildren](https://docs.unity3d.com/ScriptReference/Component.GetComponentsInChildren.html) | Returns all components of Type type in the GameObject or any of its children. |
| [GetComponentsInParent](https://docs.unity3d.com/ScriptReference/Component.GetComponentsInParent.html) | Returns all components of Type type in the GameObject or any of its parents. |
| [SendMessage](https://docs.unity3d.com/ScriptReference/Component.SendMessage.html) | Calls the method named methodName on every MonoBehaviour in this game object. |
| [SendMessageUpwards](https://docs.unity3d.com/ScriptReference/Component.SendMessageUpwards.html) | Calls the method named methodName on every MonoBehaviour in this game object and on every ancestor of the behaviour. |
| [TryGetComponent](https://docs.unity3d.com/ScriptReference/Component.TryGetComponent.html) | Gets the component of the specified type, if it exists. |
| [GetInstanceID](https://docs.unity3d.com/ScriptReference/Object.GetInstanceID.html) | Returns the instance id of the object. |
| [ToString](https://docs.unity3d.com/ScriptReference/Object.ToString.html) | Returns the name of the object. |

**Static Methods**

|  |  |
| --- | --- |
| [Destroy](https://docs.unity3d.com/ScriptReference/Object.Destroy.html) | Removes a GameObject, component or asset. |
| [DestroyImmediate](https://docs.unity3d.com/ScriptReference/Object.DestroyImmediate.html) | Destroys the object obj immediately. You are strongly recommended to use Destroy instead. |
| [DontDestroyOnLoad](https://docs.unity3d.com/ScriptReference/Object.DontDestroyOnLoad.html) | Do not destroy the target Object when loading a new Scene. |
| [FindObjectOfType](https://docs.unity3d.com/ScriptReference/Object.FindObjectOfType.html) | Returns the first active loaded object of Type type. |
| [FindObjectsOfType](https://docs.unity3d.com/ScriptReference/Object.FindObjectsOfType.html) | Returns a list of all active loaded objects of Type type. |
| [Instantiate](https://docs.unity3d.com/ScriptReference/Object.Instantiate.html) | Clones the object original and returns the clone. |

**Operators**

|  |  |
| --- | --- |
| [bool](https://docs.unity3d.com/ScriptReference/Object-operator_Object.html) | Does the object exist? |
| [operator !=](https://docs.unity3d.com/ScriptReference/Object-operator_ne.html) | Compares if two objects refer to a different object. |
| [operator ==](https://docs.unity3d.com/ScriptReference/Object-operator_eq.html) | Compares two object references to see if they refer to the same object. |

**FixedUpdate**

struct in UnityEngine.PlayerLoop

/

Implemented in:[UnityEngine.CoreModule](https://docs.unity3d.com/ScriptReference/UnityEngine.CoreModule.html)

**Description**

# Update phase in the native player loop. **UnityEngine.AIModule**

**Description**

The AI module implements the path finding features in Unity.

**Classes**

|  |  |
| --- | --- |
| [NavMesh](https://docs.unity3d.com/ScriptReference/AI.NavMesh.html) | Singleton class to access the baked NavMesh. |
| [NavMeshAgent](https://docs.unity3d.com/ScriptReference/AI.NavMeshAgent.html) | Navigation mesh agent. |
| [NavMeshBuilder](https://docs.unity3d.com/ScriptReference/AI.NavMeshBuilder.html) | Navigation mesh builder interface. |
| [NavMeshData](https://docs.unity3d.com/ScriptReference/AI.NavMeshData.html) | Contains and represents NavMesh data. |
| [NavMeshObstacle](https://docs.unity3d.com/ScriptReference/AI.NavMeshObstacle.html) | An obstacle for NavMeshAgents to avoid. |
| [NavMeshPath](https://docs.unity3d.com/ScriptReference/AI.NavMeshPath.html) | A path as calculated by the navigation system. |
| [OffMeshLink](https://docs.unity3d.com/ScriptReference/AI.OffMeshLink.html) | Link allowing movement outside the planar navigation mesh. |

**Structs**

|  |  |
| --- | --- |
| [NavMeshBuildDebugSettings](https://docs.unity3d.com/ScriptReference/AI.NavMeshBuildDebugSettings.html) | Specify which of the temporary data generated while building the NavMesh should be retained in memory after the process has completed. |
| [NavMeshBuildMarkup](https://docs.unity3d.com/ScriptReference/AI.NavMeshBuildMarkup.html) | The NavMesh build markup allows you to control how certain objects are treated during the NavMesh build process, specifically when collecting sources for building. |
| [NavMeshBuildSettings](https://docs.unity3d.com/ScriptReference/AI.NavMeshBuildSettings.html) | The NavMeshBuildSettings struct allows you to specify a collection of settings which describe the dimensions and limitations of a particular agent type. |
| [NavMeshBuildSource](https://docs.unity3d.com/ScriptReference/AI.NavMeshBuildSource.html) | The input to the NavMesh builder is a list of NavMesh build sources. |
| [NavMeshDataInstance](https://docs.unity3d.com/ScriptReference/AI.NavMeshDataInstance.html) | The instance is returned when adding NavMesh data. |
| [NavMeshHit](https://docs.unity3d.com/ScriptReference/AI.NavMeshHit.html) | Result information for NavMesh queries. |
| [NavMeshLinkData](https://docs.unity3d.com/ScriptReference/AI.NavMeshLinkData.html) | Used for runtime manipulation of links connecting polygons of the NavMesh. |
| [NavMeshLinkInstance](https://docs.unity3d.com/ScriptReference/AI.NavMeshLinkInstance.html) | An instance representing a link available for pathfinding. |
| [NavMeshLocation](https://docs.unity3d.com/ScriptReference/Experimental.AI.NavMeshLocation.html) | A world position that is guaranteed to be on the surface of the NavMesh. |
| [NavMeshQuery](https://docs.unity3d.com/ScriptReference/Experimental.AI.NavMeshQuery.html) | Object used for doing navigation operations in a NavMeshWorld. |
| [NavMeshQueryFilter](https://docs.unity3d.com/ScriptReference/AI.NavMeshQueryFilter.html) | Specifies which agent type and areas to consider when searching the NavMesh. |
| [NavMeshTriangulation](https://docs.unity3d.com/ScriptReference/AI.NavMeshTriangulation.html) | Contains data describing a triangulation of a navmesh. |
| [NavMeshWorld](https://docs.unity3d.com/ScriptReference/Experimental.AI.NavMeshWorld.html) | Assembles together a collection of NavMesh surfaces and links that are used as a whole for performing navigation operations. |
| [OffMeshLinkData](https://docs.unity3d.com/ScriptReference/AI.OffMeshLinkData.html) | State of OffMeshLink. |
| [PolygonId](https://docs.unity3d.com/ScriptReference/Experimental.AI.PolygonId.html) | Represents a compact identifier for the data of a NavMesh node. |

**Enumerations**

|  |  |
| --- | --- |
| [NavMeshBuildDebugFlags](https://docs.unity3d.com/ScriptReference/AI.NavMeshBuildDebugFlags.html) | Bitmask used for operating with debug data from the NavMesh build process. |
| [NavMeshBuildSourceShape](https://docs.unity3d.com/ScriptReference/AI.NavMeshBuildSourceShape.html) | Used with NavMeshBuildSource to define the shape for building NavMesh. |
| [NavMeshCollectGeometry](https://docs.unity3d.com/ScriptReference/AI.NavMeshCollectGeometry.html) | Used for specifying the type of geometry to collect. Used with NavMeshBuilder.CollectSources. |
| [NavMeshObstacleShape](https://docs.unity3d.com/ScriptReference/AI.NavMeshObstacleShape.html) | Shape of the obstacle. |
| [NavMeshPathStatus](https://docs.unity3d.com/ScriptReference/AI.NavMeshPathStatus.html) | Status of path. |
| [NavMeshPolyTypes](https://docs.unity3d.com/ScriptReference/Experimental.AI.NavMeshPolyTypes.html) | The types of nodes in the navigation data. |
| [ObstacleAvoidanceType](https://docs.unity3d.com/ScriptReference/AI.ObstacleAvoidanceType.html) | Level of obstacle avoidance. |
| [OffMeshLinkType](https://docs.unity3d.com/ScriptReference/AI.OffMeshLinkType.html) | Link type specifier. |
| [PathQueryStatus](https://docs.unity3d.com/ScriptReference/Experimental.AI.PathQueryStatus.html) | Bit flags representing the resulting state of NavMeshQuery operations. |

This is the C# representation of an update phase in the native player loop. It can only be used to identify the update phase in native.

[**Collider2D**](https://docs.unity3d.com/ScriptReference/Collider2D.html)**.OnTriggerEnter2D(Collider2D)**

**Parameters**

|  |  |
| --- | --- |
| **other** | The other [Collider2D](https://docs.unity3d.com/ScriptReference/Collider2D.html) involved in this collision. |

**Description**

Sent when another object enters a trigger collider attached to this object (2D physics only).

Further information about the other collider is reported in the [Collider2D](https://docs.unity3d.com/ScriptReference/Collider2D.html) parameter passed during the call.  
  
Trigger events will be sent to disabled [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)s, to allow enabling [Behaviour](https://docs.unity3d.com/ScriptReference/Behaviour.html)s in response to collisions.  
  
See Also: The [Collider2D](https://docs.unity3d.com/ScriptReference/Collider2D.html) class and the [OnTriggerExit2D](https://docs.unity3d.com/ScriptReference/Collider2D.OnTriggerExit2D.html) and [OnTriggerStay2D](https://docs.unity3d.com/ScriptReference/Collider2D.OnTriggerStay2D.html) messages.  
  
An [OnTriggerEnter2D](https://docs.unity3d.com/ScriptReference/Collider2D.OnTriggerEnter2D.html) example is shown. This example has two empty [GameObject](https://docs.unity3d.com/ScriptReference/GameObject.html)s, called GameObject1 and GameObject2. These both have script files which makes the example work. The first script, Example1, creates a [Sprite](https://docs.unity3d.com/ScriptReference/Sprite.html) and adds a [BoxCollider2D](https://docs.unity3d.com/ScriptReference/BoxCollider2D.html) and a [Rigidbody2D](https://docs.unity3d.com/ScriptReference/Rigidbody2D.html). This object falls under gravity and collides with Example2. Example2 has no visibility. (The rectangle it creates is visible in the Scene window.) Both [GameObject](https://docs.unity3d.com/ScriptReference/GameObject.html)s report the collision.  
  
The script below is Example1.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;  
  
// Create GameObject1 that falls under gravity. It will pass through

// Example2 and cause a collision. GameObject1 is moved back to

// the start position and it will again start to fall under gravity.  
  
public class Example1 : [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)

{

void Awake()

{

[SpriteRenderer](https://docs.unity3d.com/ScriptReference/SpriteRenderer.html) sr;

sr = gameObject.AddComponent<[SpriteRenderer](https://docs.unity3d.com/ScriptReference/SpriteRenderer.html)>() as [SpriteRenderer](https://docs.unity3d.com/ScriptReference/SpriteRenderer.html);

sr.color = new [Color](https://docs.unity3d.com/ScriptReference/Color.html)(0.9f, 0.9f, 0.1f, 1.0f);  
  
 [BoxCollider2D](https://docs.unity3d.com/ScriptReference/BoxCollider2D.html) bc;

bc = gameObject.AddComponent<[BoxCollider2D](https://docs.unity3d.com/ScriptReference/BoxCollider2D.html)>() as [BoxCollider2D](https://docs.unity3d.com/ScriptReference/BoxCollider2D.html);

bc.size = new [Vector2](https://docs.unity3d.com/ScriptReference/Vector2.html)(1.0f, 1.0f);  
  
 [Rigidbody2D](https://docs.unity3d.com/ScriptReference/Rigidbody2D.html) rb;

rb = gameObject.AddComponent<[Rigidbody2D](https://docs.unity3d.com/ScriptReference/Rigidbody2D.html)>() as [Rigidbody2D](https://docs.unity3d.com/ScriptReference/Rigidbody2D.html);

rb.gravityScale = 1.0f;  
  
 // A square in the [Resources](https://docs.unity3d.com/ScriptReference/Resources.html) folder is used.

gameObject.GetComponent<[SpriteRenderer](https://docs.unity3d.com/ScriptReference/SpriteRenderer.html)>().sprite = [Resources.Load](https://docs.unity3d.com/ScriptReference/Resources.Load.html)<[Sprite](https://docs.unity3d.com/ScriptReference/Sprite.html)>("square");  
  
 // GameObject1 starts 3 units in the Up direction.

gameObject.transform.position = new [Vector3](https://docs.unity3d.com/ScriptReference/Vector3.html)(0.0f, 3.0f, 0.0f);

gameObject.transform.localScale = new [Vector3](https://docs.unity3d.com/ScriptReference/Vector3.html)(0.5f, 0.5f, 1.0f);

}  
  
 private float timer = 0.0f;

private bool restart = false;  
  
 void [FixedUpdate](https://docs.unity3d.com/ScriptReference/PlayerLoop.FixedUpdate.html)()

{

if (restart == true)

{

timer = timer + [Time.deltaTime](https://docs.unity3d.com/ScriptReference/Time-deltaTime.html);

if (timer > 0.25f)

{

gameObject.transform.position = new [Vector3](https://docs.unity3d.com/ScriptReference/Vector3.html)(0.0f, 3.0f, 0.0f);

gameObject.GetComponent<[Rigidbody2D](https://docs.unity3d.com/ScriptReference/Rigidbody2D.html)>().velocity = new [Vector2](https://docs.unity3d.com/ScriptReference/Vector2.html)(0.0f, 0.0f);

restart = false;

}

}

}  
  
 // called when this [GameObject](https://docs.unity3d.com/ScriptReference/GameObject.html) collides with GameObject2.

void OnTriggerEnter2D([Collider2D](https://docs.unity3d.com/ScriptReference/Collider2D.html) col)

{

[Debug.Log](https://docs.unity3d.com/ScriptReference/Debug.Log.html)("GameObject1 collided with " + col.name);

restart = true;

timer = 0.0f;

}

}

This is Example2 which is the collide script for the second [GameObject](https://docs.unity3d.com/ScriptReference/GameObject.html):

using System.Collections;

using System.Collections.Generic;

using UnityEngine;  
  
// Create a rectangle that the other [GameObject](https://docs.unity3d.com/ScriptReference/GameObject.html) will collide with.

// Note that this [GameObject](https://docs.unity3d.com/ScriptReference/GameObject.html) has no visibility.

// (View in the [Scene](https://docs.unity3d.com/ScriptReference/SceneManagement.Scene.html) view to see this [GameObject](https://docs.unity3d.com/ScriptReference/GameObject.html).)  
  
public class Example2 : [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)

{

void Awake()

{

[BoxCollider2D](https://docs.unity3d.com/ScriptReference/BoxCollider2D.html) bc;

bc = gameObject.AddComponent<[BoxCollider2D](https://docs.unity3d.com/ScriptReference/BoxCollider2D.html)>() as [BoxCollider2D](https://docs.unity3d.com/ScriptReference/BoxCollider2D.html);

bc.size = new [Vector2](https://docs.unity3d.com/ScriptReference/Vector2.html)(3.0f, 1.0f);

bc.isTrigger = true;  
  
 gameObject.transform.localScale = new [Vector3](https://docs.unity3d.com/ScriptReference/Vector3.html)(3.0f, 1.0f, 1.0f);

gameObject.transform.position = new [Vector3](https://docs.unity3d.com/ScriptReference/Vector3.html)(0.0f, -2.0f, 0.0f);

}  
  
 void OnTriggerEnter2D([Collider2D](https://docs.unity3d.com/ScriptReference/Collider2D.html) col)

{

[Debug.Log](https://docs.unity3d.com/ScriptReference/Debug.Log.html)("GameObject2 collided with " + col.name);

}

}

[**Collider**](https://docs.unity3d.com/ScriptReference/Collider.html)**.OnCollisionEnter(Collision)**

**Parameters**

|  |  |
| --- | --- |
| **other** | The Collision data associated with this collision event. |

**Description**

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

In contrast to [OnTriggerEnter](https://docs.unity3d.com/ScriptReference/Collider.OnTriggerEnter.html), [OnCollisionEnter](https://docs.unity3d.com/ScriptReference/Collider.OnCollisionEnter.html) is passed the [Collision](https://docs.unity3d.com/ScriptReference/Collision.html) class and not a Collider. The [Collision](https://docs.unity3d.com/ScriptReference/Collision.html) class contains information, for example, about contact points and impact velocity. Notes: Collision events are only sent if one of the colliders also has a non-kinematic rigidbody attached. Collision events will be sent to disabled MonoBehaviours, to allow enabling Behaviours in response to collisions.

using UnityEngine;

using System.Collections;  
  
public class ExampleClass : [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)

{

[AudioSource](https://docs.unity3d.com/ScriptReference/AudioSource.html) audioSource;  
  
 void Start()

{

audioSource = GetComponent<[AudioSource](https://docs.unity3d.com/ScriptReference/AudioSource.html)>();

}  
  
 void OnCollisionEnter([Collision](https://docs.unity3d.com/ScriptReference/Collision.html) collision)

{

foreach ([ContactPoint](https://docs.unity3d.com/ScriptReference/ContactPoint.html) contact in collision.contacts)

{

[Debug.DrawRay](https://docs.unity3d.com/ScriptReference/Debug.DrawRay.html)(contact.point, contact.normal, [Color.white](https://docs.unity3d.com/ScriptReference/Color-white.html));

}

if (collision.relativeVelocity.magnitude > 2)

audioSource.Play();

}

}

Another example:

// A grenade

// - instantiates an explosion Prefab when hitting a surface

// - then destroys itself  
  
using UnityEngine;

using System.Collections;  
  
public class ExampleClass : [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)

{

public [Transform](https://docs.unity3d.com/ScriptReference/Transform.html) explosionPrefab;  
  
 void OnCollisionEnter([Collision](https://docs.unity3d.com/ScriptReference/Collision.html) collision)

{

[ContactPoint](https://docs.unity3d.com/ScriptReference/ContactPoint.html) contact = collision.contacts[0];

# [Quaternion](https://docs.unity3d.com/ScriptReference/Quaternion.html) rotation = [Quaternion.FromToRotation](https://docs.unity3d.com/ScriptReference/Quaternion.FromToRotation.html)([Vector3.up](https://docs.unity3d.com/ScriptReference/Vector3-up.html), [**Vector2**](https://docs.unity3d.com/ScriptReference/Vector2.html)**.ClampMagnitude**

public static [Vector2](https://docs.unity3d.com/ScriptReference/Vector2.html) **ClampMagnitude**([Vector2](https://docs.unity3d.com/ScriptReference/Vector2.html) **vector**, float **maxLength**);

**Description**

Returns a copy of vector with its magnitude clamped to maxLength.

contact.normal);

[Vector3](https://docs.unity3d.com/ScriptReference/Vector3.html) position = contact.point;

Instantiate(explosionPrefab, position, rotation);

Destroy(gameObject);

# }**GameObject**

class in UnityEngine

/

Inherits from:[Object](https://docs.unity3d.com/ScriptReference/Object.html)

/

Implemented in:[UnityEngine.CoreModule](https://docs.unity3d.com/ScriptReference/UnityEngine.CoreModule.html)

**Description**

Base class for all entities in Unity Scenes.

**Note:** Many variables in the [GameObject](https://docs.unity3d.com/ScriptReference/GameObject.html) class have been removed. To access, for example GameObject.renderer in csharp use GetComponent<Renderer>() instead.  
  
See Also: Component.

**Properties**

|  |  |
| --- | --- |
| [activeInHierarchy](https://docs.unity3d.com/ScriptReference/GameObject-activeInHierarchy.html) | Defines whether the GameObject is active in the Scene. |
| [activeSelf](https://docs.unity3d.com/ScriptReference/GameObject-activeSelf.html) | The local active state of this GameObject. (Read Only) |
| [isStatic](https://docs.unity3d.com/ScriptReference/GameObject-isStatic.html) | Gets and sets the GameObject's StaticEditorFlags. |
| [layer](https://docs.unity3d.com/ScriptReference/GameObject-layer.html) | The layer the game object is in. |
| [scene](https://docs.unity3d.com/ScriptReference/GameObject-scene.html) | Scene that the GameObject is part of. |
| [sceneCullingMask](https://docs.unity3d.com/ScriptReference/GameObject-sceneCullingMask.html) | Scene culling mask Unity uses to determine which scene to render the GameObject in. |
| [tag](https://docs.unity3d.com/ScriptReference/GameObject-tag.html) | The tag of this game object. |
| [transform](https://docs.unity3d.com/ScriptReference/GameObject-transform.html) | The Transform attached to this GameObject. |

**Constructors**

|  |  |
| --- | --- |
| [GameObject](https://docs.unity3d.com/ScriptReference/GameObject-ctor.html) | Creates a new game object, named name. |

**Public Methods**

|  |  |
| --- | --- |
| [AddComponent](https://docs.unity3d.com/ScriptReference/GameObject.AddComponent.html) | Adds a component class named className to the game object. |
| [BroadcastMessage](https://docs.unity3d.com/ScriptReference/GameObject.BroadcastMessage.html) | Calls the method named methodName on every MonoBehaviour in this game object or any of its children. |
| [CompareTag](https://docs.unity3d.com/ScriptReference/GameObject.CompareTag.html) | Is this game object tagged with tag ? |
| [GetComponent](https://docs.unity3d.com/ScriptReference/GameObject.GetComponent.html) | Returns the component of Type type if the game object has one attached, null if it doesn't. |
| [GetComponentInChildren](https://docs.unity3d.com/ScriptReference/GameObject.GetComponentInChildren.html) | Returns the component of Type type in the GameObject or any of its children using depth first search. |
| [GetComponentInParent](https://docs.unity3d.com/ScriptReference/GameObject.GetComponentInParent.html) | Retrieves the component of Type type in the GameObject or any of its parents. |
| [GetComponents](https://docs.unity3d.com/ScriptReference/GameObject.GetComponents.html) | Returns all components of Type type in the GameObject. |
| [GetComponentsInChildren](https://docs.unity3d.com/ScriptReference/GameObject.GetComponentsInChildren.html) | Returns all components of Type type in the GameObject or any of its children. |
| [GetComponentsInParent](https://docs.unity3d.com/ScriptReference/GameObject.GetComponentsInParent.html) | Returns all components of Type type in the GameObject or any of its parents. |
| [SendMessage](https://docs.unity3d.com/ScriptReference/GameObject.SendMessage.html) | Calls the method named methodName on every MonoBehaviour in this game object. |
| [SendMessageUpwards](https://docs.unity3d.com/ScriptReference/GameObject.SendMessageUpwards.html) | Calls the method named methodName on every MonoBehaviour in this game object and on every ancestor of the behaviour. |
| [SetActive](https://docs.unity3d.com/ScriptReference/GameObject.SetActive.html) | Activates/Deactivates the GameObject, depending on the given true or false value. |
| [TryGetComponent](https://docs.unity3d.com/ScriptReference/GameObject.TryGetComponent.html) | Gets the component of the specified type, if it exists. |

**Static Methods**

|  |  |
| --- | --- |
| [CreatePrimitive](https://docs.unity3d.com/ScriptReference/GameObject.CreatePrimitive.html) | Creates a game object with a primitive mesh renderer and appropriate collider. |
| [Find](https://docs.unity3d.com/ScriptReference/GameObject.Find.html) | Finds a GameObject by name and returns it. |
| [FindGameObjectsWithTag](https://docs.unity3d.com/ScriptReference/GameObject.FindGameObjectsWithTag.html) | Returns an array of active GameObjects tagged tag. Returns empty array if no GameObject was found. |
| [FindWithTag](https://docs.unity3d.com/ScriptReference/GameObject.FindWithTag.html) | Returns one active GameObject tagged tag. Returns null if no GameObject was found. |

**Inherited Members**

**Properties**

|  |  |
| --- | --- |
| [hideFlags](https://docs.unity3d.com/ScriptReference/Object-hideFlags.html) | Should the object be hidden, saved with the Scene or modifiable by the user? |
| [name](https://docs.unity3d.com/ScriptReference/Object-name.html) | The name of the object. |

**Public Methods**

|  |  |
| --- | --- |
| [GetInstanceID](https://docs.unity3d.com/ScriptReference/Object.GetInstanceID.html) | Returns the instance id of the object. |
| [ToString](https://docs.unity3d.com/ScriptReference/Object.ToString.html) | Returns the name of the object. |

**Static Methods**

|  |  |
| --- | --- |
| [Destroy](https://docs.unity3d.com/ScriptReference/Object.Destroy.html) | Removes a GameObject, component or asset. |
| [DestroyImmediate](https://docs.unity3d.com/ScriptReference/Object.DestroyImmediate.html) | Destroys the object obj immediately. You are strongly recommended to use Destroy instead. |
| [DontDestroyOnLoad](https://docs.unity3d.com/ScriptReference/Object.DontDestroyOnLoad.html) | Do not destroy the target Object when loading a new Scene. |
| [FindObjectOfType](https://docs.unity3d.com/ScriptReference/Object.FindObjectOfType.html) | Returns the first active loaded object of Type type. |
| [FindObjectsOfType](https://docs.unity3d.com/ScriptReference/Object.FindObjectsOfType.html) | Returns a list of all active loaded objects of Type type. |
| [Instantiate](https://docs.unity3d.com/ScriptReference/Object.Instantiate.html) | Clones the object original and returns the clone. |

**Operators**

|  |  |
| --- | --- |
| [bool](https://docs.unity3d.com/ScriptReference/Object-operator_Object.html) | Does the object exist? |
| [operator !=](https://docs.unity3d.com/ScriptReference/Object-operator_ne.html) | Compares if two objects refer to a different object. |
| [operator ==](https://docs.unity3d.com/ScriptReference/Object-operator_eq.html) | Compares two object references to see if they refer to the same object. |

# }[**Component**](https://docs.unity3d.com/ScriptReference/Component.html)**.GetComponent**

1public [Component](https://docs.unity3d.com/ScriptReference/Component.html) **GetComponent**(Type **type**);

**Parameters**

|  |  |
| --- | --- |
| **type** | The type of Component to retrieve. |

**Description**

Returns the component of Type type if the game object has one attached, null if it doesn't.

using UnityEngine;  
  
public class ScriptExample : [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)

{

void Start()

{

// Disable the spring on the [HingeJoint](https://docs.unity3d.com/ScriptReference/HingeJoint.html) component.

[HingeJoint](https://docs.unity3d.com/ScriptReference/HingeJoint.html) hinge = GetComponent<[HingeJoint](https://docs.unity3d.com/ScriptReference/HingeJoint.html)>();

hinge.useSpring = false;

}

}

public T **GetComponent**();

**Description**

Generic version of this method.

public [Component](https://docs.unity3d.com/ScriptReference/Component.html) **GetComponent**(string **type**);

**Description**

Returns the component with name type if the game object has one attached, null if it doesn't.

It is better to use GetComponent with a Type instead of a string for performance reasons. Sometimes you might not be able to get to the type however, for example when trying to access a C# script from Javascript. In that case you can simply access the component by name instead of type. Example:

using UnityEngine;  
  
public class ScriptExample : [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)

{

void Start()

{

// Disable the spring on the [HingeJoint](https://docs.unity3d.com/ScriptReference/HingeJoint.html) component.

[HingeJoint](https://docs.unity3d.com/ScriptReference/HingeJoint.html) hinge = GetComponent("[HingeJoint](https://docs.unity3d.com/ScriptReference/HingeJoint.html)") as [HingeJoint](https://docs.unity3d.com/ScriptReference/HingeJoint.html);

hinge.useSpring = false;

}

}

[**InputDeviceCharacteristics**](https://docs.unity3d.com/ScriptReference/XR.InputDeviceCharacteristics.html)**.Controller**

**Description**

The [InputDevice](https://docs.unity3d.com/ScriptReference/XR.InputDevice.html) is a game controller.

Game Controllers have axes and buttons that can be accessed through UnityEngine.Input.

[**Undo**](https://docs.unity3d.com/ScriptReference/Undo.html)**.IncrementCurrentGroup**

public static void **IncrementCurrentGroup**();

**Description**

Unity automatically groups undo operations by the current group index.

The current group is automatically incremented based on events eg. mouse down events, executing a menu item increments the current group. But sometimes it is necessary to manually group undo operations.

**NavMesh**

class in UnityEngine.AI

/

Implemented in:[UnityEngine.AIModule](https://docs.unity3d.com/ScriptReference/UnityEngine.AIModule.html)

**Description**

Singleton class to access the baked NavMesh.

The [NavMesh](https://docs.unity3d.com/ScriptReference/AI.NavMesh.html) is a class that can be used to do spatial queries, like pathfinding and walkability tests, set the pathfinding cost for specific area types, and to tweak global behavior of pathfinding and avoidance.  
  
In order to use the spatial queries, you need to first bake NavMesh for your Scene.  
  
See also:  
• [Building a NavMesh](https://docs.unity3d.com/Manual/nav-BuildingNavMesh.html) – for more information on how to setup and bake NavMesh  
• [Areas and Costs](https://docs.unity3d.com/Manual/nav-AreasAndCosts.html) – to learn how to use different Area types.  
• [NavMeshAgent](https://docs.unity3d.com/ScriptReference/AI.NavMeshAgent.html) – to learn how to control and move NavMesh Agents.  
• [NavMeshObstacle](https://docs.unity3d.com/ScriptReference/AI.NavMeshObstacle.html) – to learn how to control NavMesh Obstacles using scripting.  
• [OffMeshLink](https://docs.unity3d.com/ScriptReference/AI.OffMeshLink.html) – to learn how to control Off-Mesh Links using scripting.

**Static Properties**

|  |  |
| --- | --- |
| [AllAreas](https://docs.unity3d.com/ScriptReference/AI.NavMesh.AllAreas.html) | Area mask constant that includes all NavMesh areas. |
| [avoidancePredictionTime](https://docs.unity3d.com/ScriptReference/AI.NavMesh-avoidancePredictionTime.html) | Describes how far in the future the agents predict collisions for avoidance. |
| [onPreUpdate](https://docs.unity3d.com/ScriptReference/AI.NavMesh-onPreUpdate.html) | Set a function to be called before the NavMesh is updated during the frame update execution. |
| [pathfindingIterationsPerFrame](https://docs.unity3d.com/ScriptReference/AI.NavMesh-pathfindingIterationsPerFrame.html) | The maximum amount of nodes processed each frame in the asynchronous pathfinding process. |

**Static Methods**

|  |  |
| --- | --- |
| [AddLink](https://docs.unity3d.com/ScriptReference/AI.NavMesh.AddLink.html) | Adds a link to the NavMesh. The link is described by the NavMeshLinkData struct. |
| [AddNavMeshData](https://docs.unity3d.com/ScriptReference/AI.NavMesh.AddNavMeshData.html) | Adds the specified NavMeshData to the game. |
| [CalculatePath](https://docs.unity3d.com/ScriptReference/AI.NavMesh.CalculatePath.html) | Calculate a path between two points and store the resulting path. |
| [CalculateTriangulation](https://docs.unity3d.com/ScriptReference/AI.NavMesh.CalculateTriangulation.html) | Calculates triangulation of the current navmesh. |
| [CreateSettings](https://docs.unity3d.com/ScriptReference/AI.NavMesh.CreateSettings.html) | Creates and returns a new entry of NavMesh build settings available for runtime NavMesh building. |
| [FindClosestEdge](https://docs.unity3d.com/ScriptReference/AI.NavMesh.FindClosestEdge.html) | Locate the closest NavMesh edge from a point on the NavMesh. |
| [GetAreaCost](https://docs.unity3d.com/ScriptReference/AI.NavMesh.GetAreaCost.html) | Gets the cost for path finding over geometry of the area type. |
| [GetAreaFromName](https://docs.unity3d.com/ScriptReference/AI.NavMesh.GetAreaFromName.html) | Returns the area index for a named NavMesh area type. |
| [GetSettingsByID](https://docs.unity3d.com/ScriptReference/AI.NavMesh.GetSettingsByID.html) | Returns an existing entry of NavMesh build settings. |
| [GetSettingsByIndex](https://docs.unity3d.com/ScriptReference/AI.NavMesh.GetSettingsByIndex.html) | Returns an existing entry of NavMesh build settings by its ordered index. |
| [GetSettingsCount](https://docs.unity3d.com/ScriptReference/AI.NavMesh.GetSettingsCount.html) | Returns the number of registered NavMesh build settings. |
| [GetSettingsNameFromID](https://docs.unity3d.com/ScriptReference/AI.NavMesh.GetSettingsNameFromID.html) | Returns the name associated with the NavMesh build settings matching the provided agent type ID. |
| [Raycast](https://docs.unity3d.com/ScriptReference/AI.NavMesh.Raycast.html) | Trace a line between two points on the NavMesh. |
| [RemoveAllNavMeshData](https://docs.unity3d.com/ScriptReference/AI.NavMesh.RemoveAllNavMeshData.html) | Removes all NavMesh surfaces and links from the game. |
| [RemoveLink](https://docs.unity3d.com/ScriptReference/AI.NavMesh.RemoveLink.html) | Removes a link from the NavMesh. |
| [RemoveNavMeshData](https://docs.unity3d.com/ScriptReference/AI.NavMesh.RemoveNavMeshData.html) | Removes the specified NavMeshDataInstance from the game, making it unavailable for agents and queries. |
| [RemoveSettings](https://docs.unity3d.com/ScriptReference/AI.NavMesh.RemoveSettings.html) | Removes the build settings matching the agent type ID. |
| [SamplePosition](https://docs.unity3d.com/ScriptReference/AI.NavMesh.SamplePosition.html) | Finds the closest point on NavMesh within specified range. |
| [SetAreaCost](https://docs.unity3d.com/ScriptReference/AI.NavMesh.SetAreaCost.html) | Sets the cost for finding path over geometry of the area type on all agents. |

# **DelegatesCanvas**

class in UnityEngine

/

Inherits from:[Behaviour](https://docs.unity3d.com/ScriptReference/Behaviour.html)

/

Implemented in:[UnityEngine.UIModule](https://docs.unity3d.com/ScriptReference/UnityEngine.UIModule.html)

**Description**

Element that can be used for screen rendering.

Elements on a canvas are rendered AFTER Scene rendering, either from an attached camera or using overlay mode.

using System.Collections;

using System.Collections.Generic;

using [UnityEditor](https://docs.unity3d.com/ScriptReference/UnityEditor.html);

using UnityEngine;

using UnityEngine.UI;  
  
// Create a [Canvas](https://docs.unity3d.com/ScriptReference/Canvas.html) that holds a Text [GameObject](https://docs.unity3d.com/ScriptReference/GameObject.html).  
  
public class ExampleClass : [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)

{

void Start()

{

[GameObject](https://docs.unity3d.com/ScriptReference/GameObject.html) myGO;

[GameObject](https://docs.unity3d.com/ScriptReference/GameObject.html) myText;

[Canvas](https://docs.unity3d.com/ScriptReference/Canvas.html) myCanvas;

Text text;

[RectTransform](https://docs.unity3d.com/ScriptReference/RectTransform.html) rectTransform;  
  
 // [Canvas](https://docs.unity3d.com/ScriptReference/Canvas.html)

myGO = new [GameObject](https://docs.unity3d.com/ScriptReference/GameObject.html)();

myGO.name = "TestCanvas";

myGO.AddComponent<[Canvas](https://docs.unity3d.com/ScriptReference/Canvas.html)>();  
  
 myCanvas = myGO.GetComponent<[Canvas](https://docs.unity3d.com/ScriptReference/Canvas.html)>();

myCanvas.renderMode = [RenderMode.ScreenSpaceOverlay](https://docs.unity3d.com/ScriptReference/RenderMode.ScreenSpaceOverlay.html);

myGO.AddComponent<CanvasScaler>();

myGO.AddComponent<GraphicRaycaster>();  
  
 // Text

myText = new [GameObject](https://docs.unity3d.com/ScriptReference/GameObject.html)();

myText.transform.parent = myGO.transform;

myText.name = "wibble";  
  
 text = myText.AddComponent<Text>();

text.font = ([Font](https://docs.unity3d.com/ScriptReference/Font.html))[Resources.Load](https://docs.unity3d.com/ScriptReference/Resources.Load.html)("MyFont");

text.text = "wobble";

text.fontSize = 100;  
  
 // Text position

rectTransform = text.GetComponent<[RectTransform](https://docs.unity3d.com/ScriptReference/RectTransform.html)>();

rectTransform.localPosition = new [Vector3](https://docs.unity3d.com/ScriptReference/Vector3.html)(0, 0, 0);

rectTransform.sizeDelta = new [Vector2](https://docs.unity3d.com/ScriptReference/Vector2.html)(400, 200);

}

}

**Properties**

|  |  |
| --- | --- |
| [additionalShaderChannels](https://docs.unity3d.com/ScriptReference/Canvas-additionalShaderChannels.html) | Get or set the mask of additional shader channels to be used when creating the Canvas mesh. |
| [cachedSortingLayerValue](https://docs.unity3d.com/ScriptReference/Canvas-cachedSortingLayerValue.html) | Cached calculated value based upon SortingLayerID. |
| [isRootCanvas](https://docs.unity3d.com/ScriptReference/Canvas-isRootCanvas.html) | Is this the root Canvas? |
| [normalizedSortingGridSize](https://docs.unity3d.com/ScriptReference/Canvas-normalizedSortingGridSize.html) | The normalized grid size that the canvas will split the renderable area into. |
| [overridePixelPerfect](https://docs.unity3d.com/ScriptReference/Canvas-overridePixelPerfect.html) | Allows for nested canvases to override pixelPerfect settings inherited from parent canvases. |
| [overrideSorting](https://docs.unity3d.com/ScriptReference/Canvas-overrideSorting.html) | Override the sorting of canvas. |
| [pixelPerfect](https://docs.unity3d.com/ScriptReference/Canvas-pixelPerfect.html) | Force elements in the canvas to be aligned with pixels. Only applies with renderMode is Screen Space. |
| [pixelRect](https://docs.unity3d.com/ScriptReference/Canvas-pixelRect.html) | Get the render rect for the Canvas. |
| [planeDistance](https://docs.unity3d.com/ScriptReference/Canvas-planeDistance.html) | How far away from the camera is the Canvas generated. |
| [referencePixelsPerUnit](https://docs.unity3d.com/ScriptReference/Canvas-referencePixelsPerUnit.html) | The number of pixels per unit that is considered the default. |
| [renderMode](https://docs.unity3d.com/ScriptReference/Canvas-renderMode.html) | Is the Canvas in World or Overlay mode? |
| [renderOrder](https://docs.unity3d.com/ScriptReference/Canvas-renderOrder.html) | The render order in which the canvas is being emitted to the Scene. (Read Only) |
| [rootCanvas](https://docs.unity3d.com/ScriptReference/Canvas-rootCanvas.html) | Returns the Canvas closest to root, by checking through each parent and returning the last canvas found. If no other canvas is found then the canvas will return itself. |
| [scaleFactor](https://docs.unity3d.com/ScriptReference/Canvas-scaleFactor.html) | Used to scale the entire canvas, while still making it fit the screen. Only applies with renderMode is Screen Space. |
| [sortingLayerID](https://docs.unity3d.com/ScriptReference/Canvas-sortingLayerID.html) | Unique ID of the Canvas' sorting layer. |
| [sortingLayerName](https://docs.unity3d.com/ScriptReference/Canvas-sortingLayerName.html) | Name of the Canvas' sorting layer. |
| [sortingOrder](https://docs.unity3d.com/ScriptReference/Canvas-sortingOrder.html) | Canvas' order within a sorting layer. |
| [targetDisplay](https://docs.unity3d.com/ScriptReference/Canvas-targetDisplay.html) | For Overlay mode, display index on which the UI canvas will appear. |
| [worldCamera](https://docs.unity3d.com/ScriptReference/Canvas-worldCamera.html) | Camera used for sizing the Canvas when in Screen Space - Camera. Also used as the Camera that events will be sent through for a World Space [[Canvas]. |

**Static Methods**

|  |  |
| --- | --- |
| [ForceUpdateCanvases](https://docs.unity3d.com/ScriptReference/Canvas.ForceUpdateCanvases.html) | Force all canvases to update their content. |
| [GetDefaultCanvasMaterial](https://docs.unity3d.com/ScriptReference/Canvas.GetDefaultCanvasMaterial.html) | Returns the default material that can be used for rendering normal elements on the Canvas. |
| [GetETC1SupportedCanvasMaterial](https://docs.unity3d.com/ScriptReference/Canvas.GetETC1SupportedCanvasMaterial.html) | Gets or generates the ETC1 Material. |

# **EventsGameObject**

class in UnityEngine

/

Inherits from:[Object](https://docs.unity3d.com/ScriptReference/Object.html)

/

Implemented in:[UnityEngine.CoreModule](https://docs.unity3d.com/ScriptReference/UnityEngine.CoreModule.html)

**Description**

Base class for all entities in Unity Scenes.

**Note:** Many variables in the [GameObject](https://docs.unity3d.com/ScriptReference/GameObject.html) class have been removed. To access, for example GameObject.renderer in csharp use GetComponent<Renderer>() instead.  
  
See Also: Component.

**Properties**

|  |  |
| --- | --- |
| [activeInHierarchy](https://docs.unity3d.com/ScriptReference/GameObject-activeInHierarchy.html) | Defines whether the GameObject is active in the Scene. |
| [activeSelf](https://docs.unity3d.com/ScriptReference/GameObject-activeSelf.html) | The local active state of this GameObject. (Read Only) |
| [isStatic](https://docs.unity3d.com/ScriptReference/GameObject-isStatic.html) | Gets and sets the GameObject's StaticEditorFlags. |
| [layer](https://docs.unity3d.com/ScriptReference/GameObject-layer.html) | The layer the game object is in. |
| [scene](https://docs.unity3d.com/ScriptReference/GameObject-scene.html) | Scene that the GameObject is part of. |
| [sceneCullingMask](https://docs.unity3d.com/ScriptReference/GameObject-sceneCullingMask.html) | Scene culling mask Unity uses to determine which scene to render the GameObject in. |
| [tag](https://docs.unity3d.com/ScriptReference/GameObject-tag.html) | The tag of this game object. |
| [transform](https://docs.unity3d.com/ScriptReference/GameObject-transform.html) | The Transform attached to this GameObject. |

**Constructors**

|  |  |
| --- | --- |
| [GameObject](https://docs.unity3d.com/ScriptReference/GameObject-ctor.html) | Creates a new game object, named name. |

**Public Methods**

|  |  |
| --- | --- |
| [AddComponent](https://docs.unity3d.com/ScriptReference/GameObject.AddComponent.html) | Adds a component class named className to the game object. |
| [BroadcastMessage](https://docs.unity3d.com/ScriptReference/GameObject.BroadcastMessage.html) | Calls the method named methodName on every MonoBehaviour in this game object or any of its children. |
| [CompareTag](https://docs.unity3d.com/ScriptReference/GameObject.CompareTag.html) | Is this game object tagged with tag ? |
| [GetComponent](https://docs.unity3d.com/ScriptReference/GameObject.GetComponent.html) | Returns the component of Type type if the game object has one attached, null if it doesn't. |
| [GetComponentInChildren](https://docs.unity3d.com/ScriptReference/GameObject.GetComponentInChildren.html) | Returns the component of Type type in the GameObject or any of its children using depth first search. |
| [GetComponentInParent](https://docs.unity3d.com/ScriptReference/GameObject.GetComponentInParent.html) | Retrieves the component of Type type in the GameObject or any of its parents. |
| [GetComponents](https://docs.unity3d.com/ScriptReference/GameObject.GetComponents.html) | Returns all components of Type type in the GameObject. |
| [GetComponentsInChildren](https://docs.unity3d.com/ScriptReference/GameObject.GetComponentsInChildren.html) | Returns all components of Type type in the GameObject or any of its children. |
| [GetComponentsInParent](https://docs.unity3d.com/ScriptReference/GameObject.GetComponentsInParent.html) | Returns all components of Type type in the GameObject or any of its parents. |
| [SendMessage](https://docs.unity3d.com/ScriptReference/GameObject.SendMessage.html) | Calls the method named methodName on every MonoBehaviour in this game object. |
| [SendMessageUpwards](https://docs.unity3d.com/ScriptReference/GameObject.SendMessageUpwards.html) | Calls the method named methodName on every MonoBehaviour in this game object and on every ancestor of the behaviour. |
| [SetActive](https://docs.unity3d.com/ScriptReference/GameObject.SetActive.html) | Activates/Deactivates the GameObject, depending on the given true or false value. |
| [TryGetComponent](https://docs.unity3d.com/ScriptReference/GameObject.TryGetComponent.html) | Gets the component of the specified type, if it exists. |

**Static Methods**

|  |  |
| --- | --- |
| [CreatePrimitive](https://docs.unity3d.com/ScriptReference/GameObject.CreatePrimitive.html) | Creates a game object with a primitive mesh renderer and appropriate collider. |
| [Find](https://docs.unity3d.com/ScriptReference/GameObject.Find.html) | Finds a GameObject by name and returns it. |
| [FindGameObjectsWithTag](https://docs.unity3d.com/ScriptReference/GameObject.FindGameObjectsWithTag.html) | Returns an array of active GameObjects tagged tag. Returns empty array if no GameObject was found. |
| [FindWithTag](https://docs.unity3d.com/ScriptReference/GameObject.FindWithTag.html) | Returns one active GameObject tagged tag. Returns null if no GameObject was found. |

**Inherited Members**

**Properties**

|  |  |
| --- | --- |
| [hideFlags](https://docs.unity3d.com/ScriptReference/Object-hideFlags.html) | Should the object be hidden, saved with the Scene or modifiable by the user? |
| [name](https://docs.unity3d.com/ScriptReference/Object-name.html) | The name of the object. |

**Public Methods**

|  |  |
| --- | --- |
| [GetInstanceID](https://docs.unity3d.com/ScriptReference/Object.GetInstanceID.html) | Returns the instance id of the object. |
| [ToString](https://docs.unity3d.com/ScriptReference/Object.ToString.html) | Returns the name of the object. |

**Static Methods**

|  |  |
| --- | --- |
| [Destroy](https://docs.unity3d.com/ScriptReference/Object.Destroy.html) | Removes a GameObject, component or asset. |
| [DestroyImmediate](https://docs.unity3d.com/ScriptReference/Object.DestroyImmediate.html) | Destroys the object obj immediately. You are strongly recommended to use Destroy instead. |
| [DontDestroyOnLoad](https://docs.unity3d.com/ScriptReference/Object.DontDestroyOnLoad.html) | Do not destroy the target Object when loading a new Scene. |
| [FindObjectOfType](https://docs.unity3d.com/ScriptReference/Object.FindObjectOfType.html) | Returns the first active loaded object of Type type. |
| [FindObjectsOfType](https://docs.unity3d.com/ScriptReference/Object.FindObjectsOfType.html) | Returns a list of all active loaded objects of Type type. |
| [Instantiate](https://docs.unity3d.com/ScriptReference/Object.Instantiate.html) | Clones the object original and returns the clone. |

**Operators**

|  |  |
| --- | --- |
| [bool](https://docs.unity3d.com/ScriptReference/Object-operator_Object.html) | Does the object exist? |
| [operator !=](https://docs.unity3d.com/ScriptReference/Object-operator_ne.html) | Compares if two objects refer to a different object. |
| [operator ==](https://docs.unity3d.com/ScriptReference/Object-operator_eq.html) | Compares two object references to see if they refer to the same object. |

|  |  |
| --- | --- |
| [willRenderCanvases](https://docs.unity3d.com/ScriptReference/Canvas-willRenderCanvases.html) | Event that is called just before Canvas rendering happens. |

**Inherited Members**

**Properties**

|  |  |
| --- | --- |
| [enabled](https://docs.unity3d.com/ScriptReference/Behaviour-enabled.html) | Enabled Behaviours are Updated, disabled Behaviours are not. |
| [isActiveAndEnabled](https://docs.unity3d.com/ScriptReference/Behaviour-isActiveAndEnabled.html) | Has the Behaviour had active and enabled called? |
| [gameObject](https://docs.unity3d.com/ScriptReference/Component-gameObject.html) | The game object this component is attached to. A component is always attached to a game object. |
| [tag](https://docs.unity3d.com/ScriptReference/Component-tag.html) | The tag of this game object. |
| [transform](https://docs.unity3d.com/ScriptReference/Component-transform.html) | The Transform attached to this GameObject. |
| [hideFlags](https://docs.unity3d.com/ScriptReference/Object-hideFlags.html) | Should the object be hidden, saved with the Scene or modifiable by the user? |
| [name](https://docs.unity3d.com/ScriptReference/Object-name.html) | The name of the object. |

**Public Methods**

|  |  |
| --- | --- |
| [BroadcastMessage](https://docs.unity3d.com/ScriptReference/Component.BroadcastMessage.html) | Calls the method named methodName on every MonoBehaviour in this game object or any of its children. |
| [CompareTag](https://docs.unity3d.com/ScriptReference/Component.CompareTag.html) | Is this game object tagged with tag ? |
| [GetComponent](https://docs.unity3d.com/ScriptReference/Component.GetComponent.html) | Returns the component of Type type if the game object has one attached, null if it doesn't. |
| [GetComponentInChildren](https://docs.unity3d.com/ScriptReference/Component.GetComponentInChildren.html) | Returns the component of Type type in the GameObject or any of its children using depth first search. |
| [GetComponentInParent](https://docs.unity3d.com/ScriptReference/Component.GetComponentInParent.html) | Returns the component of Type type in the GameObject or any of its parents. |
| [GetComponents](https://docs.unity3d.com/ScriptReference/Component.GetComponents.html) | Returns all components of Type type in the GameObject. |
| [GetComponentsInChildren](https://docs.unity3d.com/ScriptReference/Component.GetComponentsInChildren.html) | Returns all components of Type type in the GameObject or any of its children. |
| [GetComponentsInParent](https://docs.unity3d.com/ScriptReference/Component.GetComponentsInParent.html) | Returns all components of Type type in the GameObject or any of its parents. |
| [SendMessage](https://docs.unity3d.com/ScriptReference/Component.SendMessage.html) | Calls the method named methodName on every MonoBehaviour in this game object. |
| [SendMessageUpwards](https://docs.unity3d.com/ScriptReference/Component.SendMessageUpwards.html) | Calls the method named methodName on every MonoBehaviour in this game object and on every ancestor of the behaviour. |
| [TryGetComponent](https://docs.unity3d.com/ScriptReference/Component.TryGetComponent.html) | Gets the component of the specified type, if it exists. |
| [GetInstanceID](https://docs.unity3d.com/ScriptReference/Object.GetInstanceID.html) | Returns the instance id of the object. |
| [ToString](https://docs.unity3d.com/ScriptReference/Object.ToString.html) | Returns the name of the object. |

**Static Methods**

|  |  |
| --- | --- |
| [Destroy](https://docs.unity3d.com/ScriptReference/Object.Destroy.html) | Removes a GameObject, component or asset. |
| [DestroyImmediate](https://docs.unity3d.com/ScriptReference/Object.DestroyImmediate.html) | Destroys the object obj immediately. You are strongly recommended to use Destroy instead. |
| [DontDestroyOnLoad](https://docs.unity3d.com/ScriptReference/Object.DontDestroyOnLoad.html) | Do not destroy the target Object when loading a new Scene. |
| [FindObjectOfType](https://docs.unity3d.com/ScriptReference/Object.FindObjectOfType.html) | Returns the first active loaded object of Type type. |
| [FindObjectsOfType](https://docs.unity3d.com/ScriptReference/Object.FindObjectsOfType.html) | Returns a list of all active loaded objects of Type type. |
| [Instantiate](https://docs.unity3d.com/ScriptReference/Object.Instantiate.html) | Clones the object original and returns the clone. |

**Operators**

|  |  |
| --- | --- |
| [bool](https://docs.unity3d.com/ScriptReference/Object-operator_Object.html) | Does the object exist? |
| [operator !=](https://docs.unity3d.com/ScriptReference/Object-operator_ne.html) | Compares if two objects refer to a different object. |
| [operator ==](https://docs.unity3d.com/ScriptReference/Object-operator_eq.html) | Compares two object references to see if they refer to the same object. |

|  |  |
| --- | --- |
| [OnNavMeshPreUpdate](https://docs.unity3d.com/ScriptReference/AI.NavMesh.OnNavMeshPreUpdate.html) | A delegate which can be used to register callback methods to be invoked before the NavMesh system updates. |

[**SerializedPropertyType**](https://docs.unity3d.com/ScriptReference/SerializedPropertyType.html)**.Enum**

**Description**

Enumeration property.

See Also: [SerializedProperty](https://docs.unity3d.com/ScriptReference/SerializedProperty.html).

[**GameObject**](https://docs.unity3d.com/ScriptReference/GameObject.html)**.GetComponent**

public [Component](https://docs.unity3d.com/ScriptReference/Component.html) **GetComponent**(Type **type**);

**Parameters**

|  |  |
| --- | --- |
| **type** | The type of Component to retrieve. |

**Description**

Returns the component of Type type if the game object has one attached, null if it doesn't.

Using gameObject.GetComponent will return the first component that is found and the order is undefined. If you expect there to be more than one component of the same type, use gameObject.GetComponents instead, and cycle through the returned components testing for some unique property.

using UnityEngine;  
  
public class GetComponentExample : [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)

{

void Start()

{

[HingeJoint](https://docs.unity3d.com/ScriptReference/HingeJoint.html) hinge = gameObject.GetComponent(typeof([HingeJoint](https://docs.unity3d.com/ScriptReference/HingeJoint.html))) as [HingeJoint](https://docs.unity3d.com/ScriptReference/HingeJoint.html);  
  
 if (hinge != null)

hinge.useSpring = false;

}

}

public T **GetComponent**();

**Description**

Generic version of this method.

using UnityEngine;  
  
public class GetComponentGenericExample : [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)

{

void Start()

{

[HingeJoint](https://docs.unity3d.com/ScriptReference/HingeJoint.html) hinge = gameObject.GetComponent<[HingeJoint](https://docs.unity3d.com/ScriptReference/HingeJoint.html)>();  
  
 if (hinge != null)

hinge.useSpring = false;

}

}

public [Component](https://docs.unity3d.com/ScriptReference/Component.html) **GetComponent**(string **type**);

**Parameters**

|  |  |
| --- | --- |
| **type** | The type of Component to retrieve. |

**Description**

Returns the component with name type if the game object has one attached, null if it doesn't.

It is better to use GetComponent with a Type instead of a string for performance reasons. Sometimes you might not be able to get to the type however, for example when trying to access a C# script from Javascript. In that case you can simply access the component by name instead of type.

using UnityEngine;  
  
public class GetComponentNonPerformantExample : [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)

{

void Start()

{

[HingeJoint](https://docs.unity3d.com/ScriptReference/HingeJoint.html) hinge = gameObject.GetComponent("[HingeJoint](https://docs.unity3d.com/ScriptReference/HingeJoint.html)") as [HingeJoint](https://docs.unity3d.com/ScriptReference/HingeJoint.html);  
  
 if (hinge != null)

hinge.useSpring = false;

}

}