

SOLID BEHAVIOR

View online: <https://www.construct.net/en/make-games/manuals/construct-3/behavior-reference/solid>

The **Solid behavior** makes other behaviors react to the object as if it were an impassable obstacle. Objects with this behavior are referred to as Solids. It affects the following behaviors:

- **8 Direction**, which is blocked by Solids
- **Bullet**, which can optionally bounce off Solids
- **Car**, which bounces off Solids
- **Line-of-sight**, which by default has line-of-sight obstructed by Solids
- **Move To**, which can optionally stop on solids.
- **Platform**, which can land on Solids. Platform cannot jump on to solids from underneath - for this, use the **Jump-thru** behavior.
- **Pathfinding**, which by default uses solids as path obstacles.
- **Tile movement**, which is blocked by Solids

Note that the **Physics** behavior is **not** affected by Solid objects. Instead, use the Physics behavior with *Immovable* enabled.

The Solid behavior is a fundamental attribute in Construct, and several other Construct features also interact with Solid objects. For example, the **Custom Movement** behavior has actions to push the object out of solids.

Scripting

When using JavaScript or TypeScript coding, the features of this behavior can be accessed via the **ISolidBehaviorInstance script interface**.

Avoid crushing/trapping objects with Solids

The behaviors which respond to the Solid behavior usually get stuck if Solid objects crush or otherwise trap the object deep inside the Solid object. In this case there is no solution for the movement. The only three options are 1) let the object get stuck, 2) allow the object to move inside solids, or 3) teleport the object to the nearest free space, which in some cases can be quite far away. Since options 2 and 3 can cause strange glitches if allowed, Construct will deliberately make the object unable to move, and this is the intended behavior. Therefore, it is up to you to design your games in such a way that the player cannot be crushed or trapped by moving Solid objects. You should be especially careful when moving Solids up against other Solids.

It is only by moving (or re-enabling) Solids, or using *Set position*, that objects can become trapped. If none of the Solids in your game move and you do not "teleport" the player around with *Set position*, it should be impossible for the player to ever get trapped in solids.

Solid properties

Enabled

Set whether the behavior is initially enabled or disabled. If disabled, the object no longer acts as if it is solid, and objects will be able to pass through it.

Use instance tags

Check to use the instance tags for the *Set solid collision filter* action instead of specifying tags separately in the Solid behavior's own *Tags* property. This is checked by default - the use of the Solid behavior's own *Tags* property exists only for backwards-compatibility reasons (see [Superseded features](#)).

Tags

When *Use instance tags* is unchecked, a list of tags to apply to the Solid, separated by spaces. This property exists only for backwards-compatibility reasons - enabling *Use instance tags* is recommended instead.

Solid conditions

Is enabled

True if the behavior is currently enabled. This can be changed by the *Enabled* property or the *Set enabled* action.

Solid actions

Set enabled

Enable or disable Solid for this object. Be careful not to trap objects by enabling the solid when an object is overlapping it; see *Avoid crushing/trapping objects with Solids*.

Set tags

Change the *Tags* property, affecting solid collision filtering.