BULLET BEHAVIOR

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

The **Bullet behavior** simply moves an object forwards at an angle. However, it provides extra options like gravity and bouncing that allow it to also be used like a bouncing ball. Like the name suggests it is ideal for projectiles like bullets, but it is also useful for automatically controlling other types of objects like enemies which move forwards continuously.

For two examples of the Bullet behavior see the Bouncing balls and Bouncing bullets examples in the Example Browser.

Scripting

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

Stepping bullets

Normally the bullet behavior moves a fixed distance every tick. However if it is moving extremely fast, the distance it moves in one frame can be enough to jump over obstacles in their entirety. To avoid this problem you can enable the *Step* property. This will move the object in small increments, triggering *On step* each time. In *On step* you can add an *Is overlapping* condition to check if the bullet is overlapping an obstacle with improved accuracy.

In an *On step* event the *Stop stepping* action will prevent the bullet advancing any further. This leaves the bullet in the first position it collided with an object, which is a good place to create an effect like an explosion. You can also *Destroy* the object which will also automatically stop stepping.

For an example of how stepping works, see the Bullet stepping example in the Example Browser.

While the same technique can also be used to make instant hit bullets, this is usually better done using the raycasting feature of the Line of Sight behavior, as demonstrated in the Instant hit laser example.

Bullet properties

Speed

The bullet's initial speed, in pixels per second.

Acceleration

The rate of acceleration for the bullet, in pixels per second per second. Zero will keep a constant speed, positive values accelerate, and negative values decelerate until a stop (the object will not go in to reverse).

Gravity

The force of gravity, which causes acceleration downwards, in pixels per second per second. Zero disables gravity which is useful for top-down games. Positive values cause a parabolic path as the bullet is bullet down by gravity.

Bounce off solids

If enabled, the bullet will bounce off any objects with the Solid behavior. It will also calculate the angle of reflection to bounce off realistically depending on the object's shape and angle.

Set angle

If disabled, the behavior will never change the object's angle. If enabled, the behavior always sets the object angle to the angle of motion, and if the object angle is changed, the angle of motion will be updated correspondingly.

Step

Enable stepping mode, which moves the object in small increments triggering *On step* to improve the accuracy of collisions. See *Stepping bullets* above.

Enabled

Whether the behavior is initially enabled or disabled. If disabled, it can be enabled at runtime using the *Set enabled* action.

Bullet conditions

Compare speed

Compare the current speed of the bullet, in pixels per second.

Compare distance travelled

Compare the total distance the bullet has moved since creation, in pixels. This does not take in to account altering the object position with other actions like *Set position*.

Is enabled

Test if the behavior is currently enabled. When disabled it will have no effect on the object.

On step

When *Step* is enabled, this triggers as the object moves in small increments. Add an *Is overlapping* condition to test for collisions with improved accuracy. See *Stepping mode* above for more information.

Bullet actions

Bounce off object

Make the bullet bounce off an object it has just touched. Since this requires the bullet have only just touched the object, it is generally only useful in an *On collision* event. It will also calculate the angle of reflection to bounce off realistically depending on the object's shape and angle. If the bullet is not currently overlapping the given object, or is stuck deep inside it, this action will have no effect.

Set acceleration

Set the bullet acceleration in pixels per second per second.

Set bounce off solids

Enable or disable the *Bounce off solids* property.

Set angle of motion

Set the angle the bullet is currently moving at, in degrees. *Note:* when the speed is 0, the angle of motion is always 0 and cannot be changed, since there is no motion. Therefore setting the angle of motion then the speed does not work if the object is stopped. Instead, set the speed first and then the angle of motion.

Set distance travelled

Simply changes the counter returned by the *DistanceTravelled* expression. The counter still increments according to the movement of the object.

Set enabled

Enable or disable the movement. If disabled, the behavior will stop moving the bullet, but will remember the current speed, acceleration etc. if enabled again.

Set gravity

Set the acceleration caused by gravity, in pixels per second per second.

Set speed

Set the bullet's current movement speed, in pixels per second.

Stop stepping

When *Step* is enabled, prevent the object advancing any further. This is usually done when a collision is detected with *Is overlapping*. Note that destroying the object also automatically stops stepping. For more information, see *Stepping bullets* above.

Bullet expressions

Acceleration

Get the bullet's current acceleration in pixels per second per second.

AngleOfMotion

Get the angle the bullet is currently moving at (which can be different to the object's angle), in degrees. Note when the object is stopped (with a speed of 0), the angle of motion is always 0.

DistanceTravelled

Return the total distance the bullet has moved since creation, in pixels. This does not take in to account altering the object position with other actions like *Set position*.

Gravity

Return the currently set gravity property, in pixels per second per second.

Speed

Get the bullet's current movement speed in pixels per second.