

# 8 DIRECTION BEHAVIOR

View online: <https://www.construct.net/en/make-games/manuals/construct-3/behavior-reference/8-direction>

---

The **8 Direction behavior** allows an object to be moved up, down, left, right and on diagonals, controlled by the arrow keys by default. It is often useful for controlling the player in a top-down view game. It can also be set to 4 directions or simple up/down or left/right movement which is useful for paddles or sliders. [Click here to open an example of the 8-direction behavior.](#)

The 8 Direction behavior is blocked by any objects with the [Solid behavior](#).

To set up custom or automatic controls, see the [behavior reference summary](#).

## Scripting

When using JavaScript or TypeScript coding, the features of this behavior can be accessed via the [I8DirectionBehaviorInstance script interface](#).

## 8 Direction properties

---

### Max speed

The maximum speed the object can travel at in any direction, in pixels per second.

---

### Acceleration

The rate the movement accelerates at, in pixels per second per second. When reversing against the current movement, acceleration and deceleration both contribute to slowing down.

---

### Deceleration

The rate the movement decelerates to rest when not being moved, in pixels per second per second. When reversing against the current movement, acceleration and deceleration both contribute to slowing down.

---

### Directions

Set how many directions the movement can move in. By default it is *8 Directions*, allowing movement on diagonals. *4 directions* prevents movement on diagonals, and *Up & down* or *Left & right* only allows movement along a single axis.

---

### Set angle

Whether or not the movement should also affect the objects angle. *360 degree (smooth)* will always set the object's angle to the current angle of motion. *45-degree intervals* will set the object's angle to 8 possible directions. *90-degree intervals* will set the object's angle to 4 possible directions. *No* means the behavior will not set the object's angle at all, which is useful if you want to control this yourself with events (e.g. to make the object point towards the mouse cursor).

---

### Allow sliding

If disabled, the object will simply stop when it collides with a solid. If enabled, the object will be able to continue moving along angled solids when it collides with them, essentially 'slipping' or 'sliding' along them.

---

### Default controls

If enabled, movement is controlled by the arrow keys on the keyboard. Disable to set up custom controls using the *Simulate control* action. For more information see the [behavior reference summary](#).

---

### Enabled

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

## 8 Direction conditions

---

### Allows sliding

Test if the behavior currently allows sliding along solids.

---

### Compare speed

Compare the object's current speed in pixels per second.

---

### Is enabled

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

---

### Is moving

True if the object has a non-zero speed (is not stopped). Invert to test if the object is stopped.

## 8 Direction actions

---

### Reverse

Invert the direction of motion. Useful as a simple way to bounce the object off an obstacle.

---

### **Set acceleration**

### **Set deceleration**

### **Set allow sliding**

### **Set max speed**

### **Set default controls**

These set the corresponding properties, described under *8 Direction properties*.

---

### **Set enabled**

Enable or disable the movement. If disabled, the movement no longer has any effect on the object.

---

### **Set ignoring input**

Set whether input is being ignored. If input is ignored, pressing any of the control keys has no effect. However, unlike disabling the behavior, the object can continue to move.

---

### **Set speed**

Set the current speed the object is moving at, in pixels per second.

---

### **Set vector X**

### **Set vector Y**

Set the X and Y components of the movement, in pixels per second.

---

### **Simulate control**

Simulate one of the movement controls being held down. Useful when *Default controls* is disabled. See the [behavior reference summary](#) for more information.

---

### **Stop**

A shortcut for setting the speed to zero.

## **8 Direction expressions**

---

### **Acceleration**

### **Deceleration**

### **MaxSpeed**

Return the corresponding properties, described under *8 Direction properties*.

---

### **MovingAngle**

Get the current angle of motion (which can be different to the object's angle), in degrees.

---

### **Speed**

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

---

### **VectorX**

### **VectorY**

Get the object's current speed on each axis, in pixels per second. For example, if the object is moving to the left at 100 pixels per second, *VectorX* is -100 and *VectorY* is 0.