

# PLATFORM BEHAVIOR SCRIPT INTERFACE

**View online:** <https://www.construct.net/en/make-games/manuals/construct-3/scripting/scripting-reference/behavior-interfaces/platform>

---

The `IPlatformBehaviorInstance` interface derives from `IBehaviorInstance` to add APIs specific to the Platform behavior.

## Platform behavior APIs

---

### **fallThrough()**

If the player is currently standing on a jump-thru platform, this method will make them fall through it.

---

### **resetDoubleJump(allow)**

Change whether a double-jump is allowed during the current jump. If `false` passed, a double-jump will no longer be allowed in the current jump, even if it is the first jump. If `true` passed a double-jump will be allowed again, even if a double-jump was already made in the current jump.

---

### **simulateControl(control)**

Simulate one of the movement controls being held down. Useful when `isDefaultControls` is disabled. The control is provided as a string and must be one of `"left"`, `"right"`, `"jump"`.

---

### **speed**

Get the current speed in pixels per second (read-only).

---

### **maxSpeed**

Set or get the maximum speed in pixels per second.

---

### **acceleration**

### **deceleration**

Set or get the acceleration/deceleration of the movement in pixels per second per second.

---

### **vectorX**

### **vectorY**

---

**setVector(vectorX, vectorY)****getVector()**

Set or get the X and Y components of the movement in pixels per second. The methods allow setting or getting both values at the same time.

---

**jumpStrength**

Set or get the initial vertical speed of a jump in pixels per second when the jump key is pressed.

---

**maxFallSpeed**

Set or get the maximum speed in pixels per second the object can accelerate to when in free-fall.

---

**gravity**

Set or get the acceleration caused by gravity, in pixels per second per second, at the angle given by *gravityAngle*.

---

**gravityAngle**

Set or get the angle that gravity accelerates the object, in radians. By default this points down.

---

**isDoubleJumpEnabled**

A boolean indicating if the player may make one extra mid-air jump before landing on the ground.

---

**jumpSustain**

Set or get the maximum time in seconds that the jump strength is sustained at while the jump control is being held before the effect of gravity takes over.

---

**isMoving**

A read-only boolean indicating if the object is currently moving. This checks if either `vectorX` or `vectorY` are non-zero.

---

**isOnFloor**

A read-only boolean indicating if the object is currently standing on a solid or jump-thru.

---

**isByWall(side)**

Test if a solid blocking horizontal movement is immediately to the object's left or right, returning a boolean indicating if a wall was found. Jump-thrus do not count as walls. Pass either `"left"` or `"right"` as the *side*.

---

**isJumping**

A read-only boolean indicating if the object is currently moving upwards.

---

**isFalling**

A read-only boolean indicating if the object is currently in free-fall.

---

**ceilingCollisionMode**

A string of either `"stop"` (the default) or `"preserve-momentum"` (keep the vertical speed) indicating how to handle ceiling collisions.

---

**isDefaultControls**

A boolean indicating if the default controls (using the arrow keys) are enabled.

---

**isIgnoringInput**

A boolean indicating if input is currently 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.

---

**isEnabled**

A boolean indicating if the behavior is enabled. If disabled, the behavior no longer has any effect on the object.