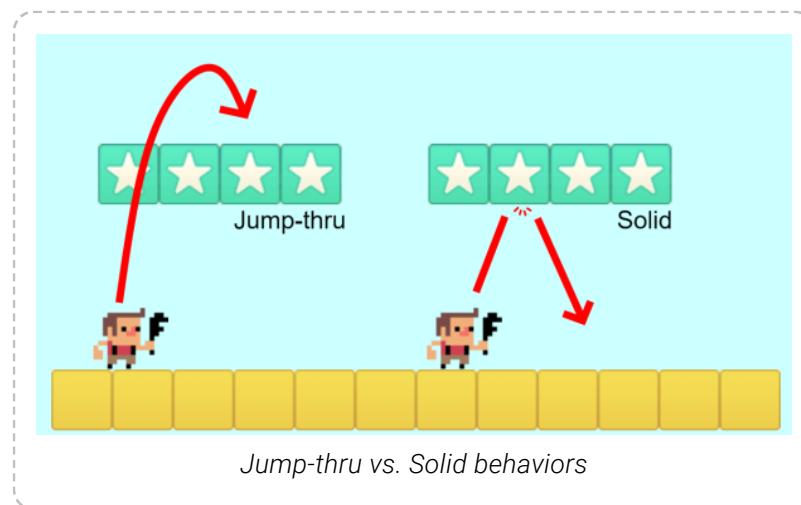


PLATFORM BEHAVIOR

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

The **Platform behavior** implements a side-view "jump and run" style movement. It supports slopes, moving platforms, "jump-thru" platforms, and arbitrary angles of gravity. There are several examples of the Platform behavior that can be found in the [Start Page](#).

The Platform behavior will land on any objects with the **Solid** or **Jump-thru** behaviors. Jump-thru is different in that the Platform movement can jump on to a Jump-thru from underneath, whereas jumping in to a solid from underneath causes the player to bounce off. The image below illustrates the difference.



By default the Platform movement is controlled by the left and right arrow keys and up arrow to jump. To set up custom or automatic controls, see the [behavior reference summary](#).

Scripting

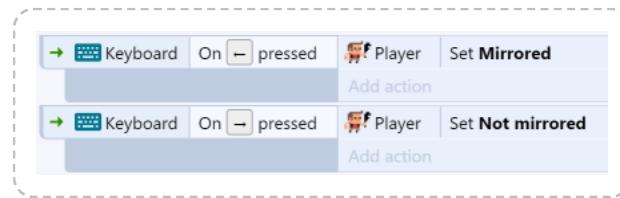
When using JavaScript or TypeScript coding, the features of this behavior can be accessed via the [IPlatformBehaviorInstance](#) script interface.

Implementing reliable platform movements

For the most reliable platform movement, it is recommended to use a **invisible rectangle sprite with no animations** with the Platform movement. Then, the animated player object can be positioned on top of that. Otherwise, the changing size and collision polygon of the object as its animation plays can interfere with the Platform movement's floor, wall and slope detection, causing a shaking or glitchy movement. For more information and a complete guide, see the tutorial [How to make a Platform game](#).

Mirroring the player

It is not necessary to duplicate all your artwork for the player for facing both left and right. Instead, simply draw all the player's artwork for facing to the right, and use the Sprite object's Set mirrored action to mirror the player's image. Set the player mirrored when pressing the movement controls. An example is shown below.



A common question is how to make the player shoot left and right, since even when mirrored the player will still shoot objects to the right. To solve this, use the *Is mirrored* condition, and if it is true, shoot to the left instead.

Platform properties

Max speed

The maximum floor speed in pixels per second.

Acceleration

The horizontal movement acceleration in pixels per second per second.

Deceleration

The horizontal movement deceleration in pixels per second per second. When moving in the opposite direction to the direction of motion, acceleration and deceleration combine.

Jump strength

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

Gravity

The acceleration caused by gravity, in pixels per second per second.

Max fall speed

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

Double jump

If enabled, the player may make one extra mid-air jump before landing on the ground.

Jump sustain

Maximum time in milliseconds that the jump strength is sustained at while the jump control is being held before the effect of gravity takes over. This allows for variable height jumps depending on whether the jump control is tapped or held. For example if set to 200, then the jump velocity is sustained for up to the first 200ms of holding the jump control.

Default controls

If enabled, movement is controlled by the left and right arrow keys and the up arrow key to jump. 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.

Platform conditions

Compare speed

Compare the current speed of the object in pixels per second.

Is by wall

Test if a solid blocking horizontal movement is immediately to the object's left or right. Jump-thrus do not count as walls.

Is double-jump enabled

True if double jumps are currently enabled. This is set by the *Double jump* property or Set *double-jump* action.

Is enabled

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

Is falling

True if the object is in free-fall.

Is jumping

True if the object is moving upwards.

Is moving

True if the object's speed is non-zero.

Is on floor

True if the object is currently standing on a solid or jump-thru.

On fall

On jump

On landed

On moved

On stopped

These are *animation triggers*, which trigger when the platform movement is moving in to each state. If your object has animations for any of these states, you should set the appropriate animation in each trigger. This helps save you implementing the logic to detect state transitions yourself.

Platform actions

Fall through

If the player is currently standing on a [jump-thru](#) platform, this action will make them fall through it. This is useful for adding an additional control, e.g. down arrow, to jump down from jump-thru platforms.

Reset double jump

Change whether a double-jump is allowed during the current jump. If disabled a double-jump will no longer be allowed in the current jump, even if it is the first jump; if enabled a double-jump will be allowed again, even if a double-jump was already made in the current jump.

Set acceleration

Set deceleration

Set default controls

Set double-jump

Set gravity

Set jump strength

Set jump sustain

Set max fall speed

Set max speed

Set the corresponding properties. For more information, see *Platform properties*.

Set angle of gravity

Change the angle of gravity, in degrees. By default it is 90 (downwards on the screen). This can interact interestingly with layer rotation.

Set ceiling collision

Change how the behavior handles collisions with the ceiling, when the movement is jumping up and hits a solid. The default mode is *Stop* which means the vertical momentum is immediately set to zero so the movement immediately falls down again. Changing the mode to *Preserve momentum* will not change the vertical momentum when hitting a solid while jumping. This means the player can stick to the ceiling until gravity overcomes their upwards momentum; it can also allow the player to jump up past a corner of a solid even if they hit the solid vertically first.

Set enabled

Enable or disable the Platform movement. When disabled, the behavior has no effect on the object at all.

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, e.g. if in free-fall.

Set vector X

Set vector Y

Manually set the horizontal and vertical components of motion, in pixels per second. For example, setting the vector Y to -1000 would cause a jump with strength 1000, which could be useful for implementing springs.

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.

Platform expressions

Acceleration

Deceleration

Gravity

JumpStrength

JumpSustain

MaxFallSpeed**MaxSpeed**

Return the corresponding properties. For more information, see *Platform properties*.

GravityAngle

Get the current angle of gravity, in degrees.

MovingAngle

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

Speed

Return the current overall speed in pixels per second.

VectorX**VectorY**

Return the current X and Y components of motion, in pixels per second.