

TILE MOVEMENT BEHAVIOR SCRIPT INTERFACE

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

The `ITileMovementBehaviourInstance` interface derives from `IBehaviorInstance` to add APIs specific to the **Tile Movement behavior**.

Tile Movement behavior APIs

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.

isDefaultControls

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

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"`, `"up"`, `"down"`.

isEnabled

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

setSpeed(x, y)

Modify the speed property of the behavior. X and Y are numerical values that define the speed on the respective axes.

getSpeed()

Read the speed property of the behavior. The speed is returned as an array with 2 numerical elements, representing the speed for the X and Y axes respectively.

setGridPosition(x, y, immediate)

Modify the target grid position of the behavior. X and Y are numerical values that define the new target in grid space. Immediate is a boolean flag that indicates if the instance should immediately move to the target position, or move there as if the user was controlling the instance.

getGridPosition()

Read the current instance position in grid space. Returns an array of 2 numerical elements, being the position in the X and Y axes respectively.

modifyGridDimensions(width, height, xOffset, yOffset)

Redefine the dimensions of the grid. The Width and height parameters are numbers that specify the new size for the grid cells. The xOffset and yOffset parameters are numbers that specify the grid's offset from the world space. If xOffset and yOffset are a multiple of width or height respectively they will have no effect on the alignment of the grid, but will change the grid space position of each cell.

isMoving()

Returns a boolean indicating if the instance is currently trying to move to a new target position

isMovingDirection(direction)

Returns a boolean indicating if the instance is currently trying to move to a new target position, in the given direction. Useful for deciding which animation to show for your character. The direction parameter is a string, which must be one of the following `"left"`, `"right"`, `"up"`, `"down"`.

canMoveto(x, y)

Allows you to check if the instance would collide with an object if it was moved to the grid cell specified by the numerical parameters x and y. The parameters being the co-ordinates of the cell in grid space. This does not check for obstructions between the current position and the target position.

canMoveDirection(direction, distance)

Allows you to check if the instance would collide with an object if it moved a number of cells in a given direction. The direction is specified by the parameter direction, which must be a string of the value `"up"`, `"left"`, `"right"` or `"down"`. The distance is specified by the parameter distance, which must be a number indicating the number of cells to travel (not the distance in world space). This will check for any obstructions between the current position and the target. This is quite useful for character AI.

getPosition()

Returns the current target position in world space as an array of 2 numerical elements. These elements being the X and Y positions respectively.

getGridTargetPosition()

Returns the current target position in grid space as an array of 2 numerical elements. These elements being the X and Y positions respectively.

toGridSpace(x, y)

Convert a given co-ordinate in world space, to grid space. The return value is an array of 2 numerical elements. These elements being the X and Y positions in grid space respectively.

fromGridSpace(x, y)

Convert a given co-ordinate in grid space, to world space. The return value is an array of 2 numerical elements. These elements being the X and Y positions in world space respectively.