

LWScratch


by loosewheel

Version 0.1.0

This mod provides scratch programmable robots.



The first time a robot is placed in the world a form opens asking the player that placed it, if the machine is public or private. If private is selected, the player becomes the owner and other players (except those with *protection_bypass* privilege) cannot access it.

The persistence button  toggles on and off. If persistence is on, the block the robot is in remains loaded when out of range. This persistence is retained across world startups. Robots retain their persistence state when moved. The maximum force loaded blocks is limited to the *max_forceloaded_blocks* setting (default is 16).

Each robot can be given a name, by entering the name in the *name* field and clicking the *Set* button. The name will display when the robot is pointed at or as the tool tip if it is in an inventory.

Each robot has a storage area (bottom left).


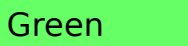
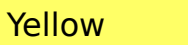
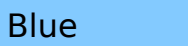

While a robot is running *sneak + punch* will open a form to stop it.

Robots are programmed graphically, by dragging a command from a pallet (top left) to the program sheet (top right). Items can be dragged from the inventories. These are only markers, the item is not used. To remove an item from the program sheet, drag it to an empty space in the command pallet. To clear the whole program click the clear button. Commands are run in order, left to right per line, then down the lines.

Block delimiting (for loop and if) is by indenting. When a following line is indented to the same level or less, this marks the end of the block.

To run the program click the power  button . If the program has an error a red message below the program sheet details the error.

Command items are color coded by type:

	Statement, controls program flow.
	Value, represents a given value.
	Condition, results as true or false.
	Action, performs an action of some kind.
	Sheet action, used to edit the program sheet.

Statements



Loop

Followed by a condition which evaluates to true or false. The following lines of commands indented greater than the loop statement will run repeatedly until the condition is false.

Each loop has an internal counter, which starts at zero and increments by one every iteration.



If

Followed by a condition which evaluates to true or false. The following lines of commands indented greater than the if statement will run once if the condition is true.

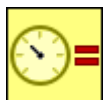
Values



Number

Can be set with an number value when placed (or moved) in the program sheet. An input field and button labelled *Number* appear. Set the value and click the button to set the value. Hovering over the number item the tool tip displays its current value.

Conditions



Counter is equal to *Outside of a loop the counter is always zero.*



Followed by a number item, and results in true if the loop's counter is less than the number.

Counter is less than *Outside of a loop the counter is always zero.*



Followed by a number item, and results in true if the loop's counter is greater than the number.

Counter is greater than *Outside of a loop the counter is always zero.*



If followed by an inventory item, true if the node in the up direction matches the inventory item.

Detect up If followed by a blank space, true if there is any node in the up direction.



If followed by an inventory item, true if the node in the down direction matches the inventory item.

Detect down If followed by a blank space, true if there is any node in the down direction.



If followed by an inventory item, true if the node in the forward direction matches the inventory item.

Detect forward If followed by a blank space, true if there is any node in the forward direction.



If followed by an inventory item, true if the node in the forward, up direction matches the inventory item.

Detect forward up If followed by a blank space, true if there is any node in the forward, up direction.



If followed by an inventory item, true if the node in the forward, down direction matches the inventory item.

Detect forward down If followed by a blank space, true if there is any node in the forward, down direction.



If followed by an inventory item, true if the node in the back direction matches the inventory item.

Detect back If followed by a blank space, true if there is any node in the back direction.



If followed by an inventory item, true if the node in the back, up direction matches the inventory item.

Detect back up If followed by a blank space, true if there is any node in the back, up direction.



If followed by an inventory item, true if the node in the back, down direction matches the inventory item.

Detect back down If followed by a blank space, true if there is any node in the back, down direction.



Contains item

If followed by an inventory item, true if the robot's storage contains at least one of the inventory item.



Item fits

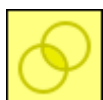
If followed by an inventory item, true if one of the inventory item can fit in the robot's storage.

If followed by a blank space, true if the robot's storage has at least one empty slot (can fit anything).



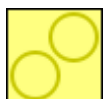
Not

Inverts the next condition result (true to false, or false to true).



And

Placed between two conditions and is true only if both the left and right conditions are true.



Or

Placed between two conditions and is true if either the left or right (or both) condition is true.

Actions



Move forward

Moves one node in the forward direction, if nothing is there.



Move backward

Moves one node in the backward direction, if nothing is there.



Move up

Moves one node in the up direction, if nothing is there.



Move down

Moves one node in the down direction, if nothing is there.



Turn left

Turns 90 degrees in the left direction.



Turn right

Turns 90 degrees in the right direction.



Dig up

Digs the node in the up direction, if there is anything there. If dug, node is placed in the robot's storage if there is room, otherwise it is dropped.



Dig down

Digs the node in the down direction, if there is anything there. If dug, node is placed in the robot's storage if there is room, otherwise it is dropped.



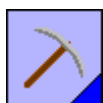
Dig forward

Digs the node in the forward direction, if there is anything there. If dug, node is placed in the robot's storage if there is room, otherwise it is dropped.



Dig forward up

Digs the node in the forward up direction, if there is anything there. If dug, node is placed in the robot's storage if there is room, otherwise it is dropped.



Dig forward down

Digs the node in the forward down direction, if there is anything there. If dug, node is placed in the robot's storage if there is room, otherwise it is dropped.



Dig back

Digs the node in the back direction, if there is anything there. If dug, node is placed in the robot's storage if there is room, otherwise it is dropped.



Dig back up

Digs the node in the back up direction, if there is anything there. If dug, node is placed in the robot's storage if there is room, otherwise it is dropped.



Dig back down

Digs the node in the back down direction, if there is anything there. If dug, node is placed in the robot's storage if there is room, otherwise it is dropped.



Place up

Followed by an inventory item. Places the given inventory item in the up direction if there is nothing at that position. The item must be in the robot's storage.



Place down

Followed by an inventory item. Places the given inventory item in the down direction if there is nothing at that position. The item must be in the robot's storage.



Place forward

Followed by an inventory item. Places the given inventory item in the forward direction if there is nothing at that position. The item must be in the robot's storage.



Place forward up

Followed by an inventory item. Places the given inventory item in the forward, up direction if there is nothing at that position. The item must be in the robot's storage.



Place forward down

Followed by an inventory item. Places the given inventory item in the forward, down direction if there is nothing at that position. The item must be in the robot's storage.



Place back

Followed by an inventory item. Places the given inventory item in the back direction if there is nothing at that position. The item must be in the robot's storage.



Place back up

Followed by an inventory item. Places the given inventory item in the back, up direction if there is nothing at that position. The item must be in the robot's storage.



Place back down

Followed by an inventory item. Places the given inventory item in the back, down direction if there is nothing at that position. The item must be in the robot's storage.



Pull

If followed by an inventory item, moves one of the given inventory items from an inventory (chest) immediately in front of the robot, into the robot's storage if it can fit.

If followed by a blank space, moves everything from an inventory (chest) immediately in front of the robot, into the robot's storage or as much as can fit.



Put

If followed by an inventory item, moves one of the given inventory items from the robot's storage into an inventory (chest) immediately in front of the robot, if it can fit.

If followed by a blank space, moves everything from the robot's storage into an inventory (chest) immediately in front of the robot, or as much as can fit.



Drop

If followed by an inventory item, drops one of the given inventory items from the robot's storage into the world, if it contains one.

If followed by a blank space, drops everything from the robot's storage into the world.



Trash

If followed by an inventory item, destroys (gone forever) one of the given inventory items in the robot's storage, if it contains one.

If followed by a blank space, destroys (gone forever) everything in the robot's storage.



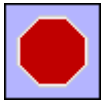
Craft

Followed by an inventory item. Crafts the given inventory item. The materials for the craft must be in the robot's storage.



Wait

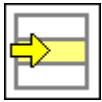
Followed by a number item. Pauses the robot's program by the number value in tenths of a second (10 = 1 second pause).



Stop

Stops the robot's program.

Sheet actions



Insert line

Inserts a line in the program sheet where it is dropped. The last line of the sheet is lost.



Remove line

Removes a line in the program sheet where it is dropped. The removed line of the sheet is lost.