LWScratch

by loosewheel

Version 0.2.1

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 *Robot* 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 (center right).

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 right) to the program sheet (left). 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:

Orange	Statement, controls program flow.
Green	Value, represents (contains) a value.
Yellow	Condition, results as true or false.
Blue	Action, performs an action of some kind.
White	Sheet action, used to edit the program sheet.

Working with variables

All variable items, whether they are values, conditions or actions, must be given a name. Set the name by placing it in the top slot, entering the name in the *Value* field and clicking *Set*. All variable items with the same name are the same variable value, whether being set with a value, testing or using its value. Variables are never given a value in the top slot, only a name. Their value is always set or changed in the program. The Number and Text commands hold values.

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 a number value. To set the value, place it in the value slot at the top, enter the desired value in the *Value* field and click the *Set* button. Hovering over the number item, the tool tip displays its current value.

Text

Can be set with a text value. To set the value, place it in the value slot at the top, enter the desired value in the *Value* field and click the *Set* button. Hovering over the text item, the tool tip displays its current value.

Variable

Can be set with a name. To set the name, place it in the value slot at the top, enter the desired name in the *Value* field and click the *Set* button. Hovering over the variable item, the tool tip displays its current name.
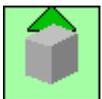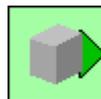
|							
Name up	Name down	Name forward	Name forward up	Name forward down	Name back	Name back up	Name back down
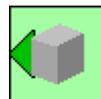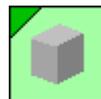
Is the name of the node in the relevant direction. If no node is there it is blank text.

Conditions
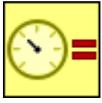
Counter is
equal to

Followed by a number or variable item, and results in true if the loop's counter is equal to the number.

Outside of a loop the counter is always zero.

Counter is
less than

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

Outside of a loop the counter is always zero.

Counter is
greater
than

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

*Outside of a loop the counter is always zero.*

Counter is
even

Results in true if the counter is currently an even number.

Counter is
odd

Results in true if the counter is currently an odd number.
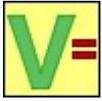
Variable is
equal to

If followed by a number, text, variable or name item, results in true if the variable is equal to the following value.

If followed by inventory item, results in true if the variables value equals the item's name.

Must be set with a name in the top slot.

Variable is
less than

Followed by a number or variable, and results in true if the variable is less than the following value.

Must be set with a name in the top slot.

Variable is
greater
than

Followed by a number or variable, and results in true if the variable is greater than the following value.

Must be set with a name in the top slot.

Variable is
even

Results in true if the variable is currently an even number (integer part).

Must be set with a name in the top slot.

Variable is
odd

Results in true if the variable is currently an odd number (integer part).

Must be set with a name in the top slot.

| Detect up | Detect down | Detect forward | Detect forward up | Detect forward down | Detect back | Detect back up | Detect back down |

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

If followed by a text or variable item, true if the node in the relevant direction match the text or variable's value.

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

Contains
item

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

If followed by a text or variable item, true if robot's storage contains at least one of the inventory items named in the text or variable's value.

If followed by a blank space, true if the robot's storage contains anything at all.

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 text or variable item, true if one of the inventory item named in the text or variable's value 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.
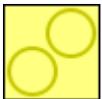
Or

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

Actions
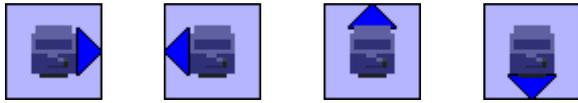
Move forward	Move backward	Move up	Move down

Moves one node in the relevant direction, if nothing is there.
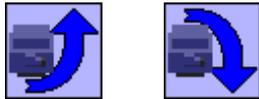
Turn left	Turn right

Turns the robot 90 degrees in the relevant direction.
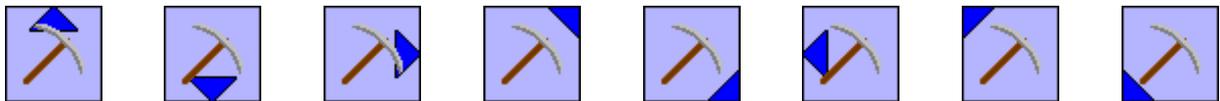
Dig up	Dig down	Dig forward	Dig forward up	Dig forward down	Dig back	Dig back up	Dig back down

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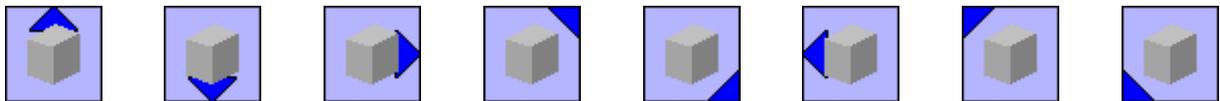
Place up	Place down	Place forward	Place forward up	Place forward down	Place back	Place back up	Place back down

If followed by an inventory item, places the given inventory item in the relevant direction if there is nothing at that position.

If followed by a text or variable item, places the inventory item named in the text or variable's value in the relevant direction if there is nothing at that position.

The item must be in the robot's storage.
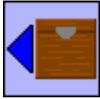
Pull

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 text or variable item, moves one of the inventory items named in the text or variable's value 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.

Pull stack

If followed by an inventory item, moves up to a full stack 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 text or variable item, moves up to a full stack of the inventory items named in the text or variable's value from an inventory (chest) immediately in front of the robot, into the robot's storage if it can fit.
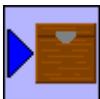
Put

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 text or variable item, moves one of the inventory items named in the text or variable's value 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.

Put stack

If followed by an inventory item, moves up to a full stack 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 text or variable item, moves up to a full stack of the inventory items named in the text or variable's value from the robot's storage into an inventory (chest) immediately in front of the robot, if it 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 text or variable item, drops one of the inventory items named in the text or variable's value 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.

Drop stack

If followed by an inventory item, drops up to a full stack of the given inventory items from the robot's storage into the world, if it contains any.

If followed by a text or variable item, drops up to a full stack of the inventory items named in the text or variable's value from the robot's storage into the world, if it contains any.

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 text or variable item, destroys (gone forever) one of the inventory items named in the text or variable's value in the robot's storage, if it contains one.

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

Trash stack

If followed by an inventory item, destroys (gone forever) up to a full stack of the given inventory items in the robot's storage, if it contains any.

If followed by a text or variable item, destroys (gone forever) up to a full stack of the inventory items named in the text or variable's value in the robot's storage, if it contains any.

Craft

If followed by an inventory item, crafts the given inventory item. The materials for the craft must be in the robot's storage.

If followed by a text or variable item, crafts the inventory item named in the text or variable's value. The materials for the craft must be in the robot's storage.

Wait

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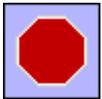

Stop

Stops the robot's program.

Chat

Followed by a text or variable item, the contents of which is sent to the chat. If the robot is private the message is only sent to the owner. If the robot is public and the *Allow public chat* setting is enabled the message is sent to all.
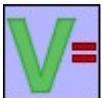

Variable
assign

If followed by a name, assigns the node name in the given direction to this variable.

If followed by a number, text or variable, assigns the value in the following number, text or variable to this variable.

If followed by inventory item, assigns the name of the inventory item to this variable.

Must be set with a name in the top slot.
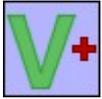
Variable
add

If followed by a name, adds the node name in the given direction to the end of this variable's current value.

If followed by a number, text or variable. If either of the values are text, adds the following value to the end of this variable's current value. Otherwise adds, as numbers, the   two values and assigns the result to this variable.

Must be set with a name in the top slot.

Variable
minus

Followed by a number or variable. Subtracts the following value from this variables value and assigns the result to this variable.

Must be set with a name in the top slot.
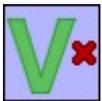
Variable
multiply

Followed by a number or variable. Multiplies the following value with this variables value and assigns the result to this variable.

Must be set with a name in the top slot.
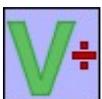
Variable
divide

Followed by a number or variable. Divides this variables value by the following value and assigns the result to this variable.

Must be set with a name in the top slot.

Sheet Actions
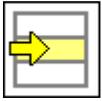
 Insert line

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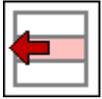
 Remove
  line

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