Scheme functions specific to Scheme–Man

(Revision of July 23, 2025)

Generic macros

`(repeat n`
`  body ...)`
	Evaluates `body` *n* times.

`(inc! x)`	Increments the value of `x`. `x` must be a variable.
`(dec! x)`	Decrements the value of `x`. `x` must be a variable.
`(push! list x)`	Adds the value `x` at the front of `list`. `list` must be a variable.
`(pop! list)`	Extracts and returns the element at the front of `list`. `list` must be a variable.

Status messages

`(status format ...)`	Shows the formatted message on top of the player's head. `format` can be a string or something with more structure: `(status "I am Scheme-Man!")` or `(status "sum: ~a, not: ~a" (+ 1 2 3) (not #t))`.
`(clear-status)`	Equivalent to `(status "")`.

Checking and resetting the state of the game

`(playing?)`	Returns `#f` only when the player has lost.
`(reset-level)`	Resets the level the way it was first loaded but does not the state of the Scheme interpreter (all the definitions are kept). This can also be done while the window is focused by pressing `Ctrl-R`.

Walking

`(walk)`
`(walk n)`
	Walks one or *n* steps in the direction the player is facing.

`(walk-while predicate)`	Walks while `predicate` evaluates to true. For example, `(walk-while (see? 'coin))` walks while the player sees a coin in their line of sight.

`(go-back)`
`(go-back n)`
	Walks one or *n* steps backwards from the direction the player is facing.

Turning

`(turn dir)`	Turns in the specified direction. The direction must be one of the following symbols: left, right or opposite. Example: `(turn 'right)`.

Seeing	

`(see)`
`(see distance)`
	Returns a symbol with the name of the object that's right in front or at a specified `distance` from the player. `distance` must be a strictly positive number. Example: `(see 2)` could return `floor` or `coin`.
	
	Returns an empty list if at that position there is no floor.
	
	If there is no floor between two patches of land, you will not be able to see and the return value will always be an empty list.

`(see? object)`
`(see? object min-dist)`
	Returns `#t` if an object of type `object` is found at a distance at least one tile in the first case or `min-dist` tiles in the second. Otherwise, this function returns `#f`.

`(can-walk?)`	Returns `#t` if there is floor in front of the player for them to walk to.

Remembering and going back to places	
`(remember-place name)`	Remembers the player's current location and gives it a `name`.
`(go-back-to place-name)`	Returns to a previously remembered place.

`(with-route-backwards`
`  body ...)`
	Perform a series of actions in `body` and then return to the place the player was before executing those actions.