

Maze walker commands

1. Commands

- a. Forward (fd)
 - b. Turn-left (tl)
 - c. Turn-right (tr)
 - d. Turn-back (tb)
 - e. pick-one-randomly
 - i. returns one **command** of
 1. Forward, turn-left, turn-right, random turn- L/Forward , random turn- R/Forward, random turn-L/turn-R/Forward, turn-back
2. Check-for-turns (cft)
- a. Returns one **decision-condition** of
 - i. Straight (s), left-turn (l), right-turn (r), left-and-right (lr), left-and-straight (ls), right-and-straight (rs), left-right-straight (lrs), none (n)
3. repeat-until-target
4. if **decision-condition** then **command**
- a. e.g., if-lt tl ; if left turn is possible (maybe other turns are possible too), then turn left
 - b. if-lto tl ; if left turn only is possible (no other options), then turn left

Maze walker programs

Random walk – not implemented yet

Repeat-until-target

Forward

Check-for-turns

If-can-turn left-turn then turn-left

If-can-turn right-turn then turn-right

If-can-turn left-and-right then pick-one-randomly

If-can-turn left-and-straight then pick-one-randomly

If-can-turn right-and-straight then pick-one-randomly

If-can-turn left-right-straight then pick-one-randomly

If decision-wall then turn-back

Hand-on-wall walk (say, left hand) – **long version** = precise control over turns (you can decide which way to turn for every junction type/option). Since it is a left hand walk, turning right is the last option/command

Repeat-until-target

Forward

Check-for-turns

If-can-turn left-turn-only then turn-left

If-can-turn right-turn-only then turn-right

If-can-turn left-and-straight then turn-left

If-can-turn left-and-right then turn-left

If-can-turn left-right-straight then turn-left

If-can-turn right-and-straight then stay straight

If-can-turn none then turn-back

Or in **short-hand notation** for left-hand-on-wall walk:

repeat-until-target

fd

cft

if-lto tl

if-rto tr

if-slo tl

if-lro tl

if-slrl tl

if-sro ss

if-n tb

Maze condition	Possible actions	
S (only)	ss	Stay straight
N (only)	tb	Turn back
L (only)	tl	Turn left
R (only)	tr	Turn right
S+l	Ss, tl	
S+r	Ss, tr	
S+l+r	Ss, tl, tr	
L+r	Tl, tr	

Hand-on-wall walk (say, **left hand**) – **short version** (less fine control of the walker actions) – if you can turn left, then turn left, if you can go straight or turn right then go straight (so prefer left every time; turn right only if absolutely necessary). Order of commands is very important!

for a **left-hand** walk:

repeat-until-target

fd

cft

if-lt tl ; if left turn (and maybe other options, doesn't matter), turn left

if-st ss ; if straight or any turn, stay straight

if-rt tr ; if right turn (and maybe other options, doesn't matter), turn right

if-n tb

for a **right-hand** walk:

repeat-until-target

fd

cft

if-rt tr ; if right turn (and maybe other options, doesn't matter), turn right

if-st ss ; if straight or any turn, stay straight

if-lt tl ; if left turn (and maybe other options, doesn't matter), turn left

if-n tb

To walk a maze while marking a line to indicate where the walker has already been:

If you want to favor **left-hand-walk**, in case of no marks at a junction, or equal number of marks in all directions at a junction:

Repeat-until-target

fd

cft ; check for turns – will ALSO check for marked path counts.

; check for marks. Returns values: l0 or l1 or l2, r0, or r1 or r2, s0, s1, s2, l0r0, l1r0s2, etc.

If-lmse tl ; we want left-hand walk, so favor/start left: If-mark-left-smallest-or-equal turn left

If-mrse tr ; If-mark-right-smallest-or-equal turn right

If-msse ss ; if-mark-straight-smallest-or-equal stay straight