SensingHw	MovingHwE	MovingHw
⚡ neighbours: nat	⚡ obstacle: Position	ⓞ move(l: real, a: real)

Movement

Ⓡ MovingHw
π lv: real, av: real, MB: real, alpha: nat, K: nat
✖ n: nat, p: Position, turned: boolean
🕘 MBC
ⓘ MovingHwE
ⓘ SensingHw

obstacle: Position ☐

neighbours: nat ☐

ⓘ #MBC

MovementAndAvoidance

obstacle?p[since(MBC)<(MB-360/av)]

Move	Avoid
entry move(lv, 0)	entry if (p==Position::left) then move(0, av) else move(0, -av) end; wait ([0,floor((360/av)/K)])

[since(MBC)>=MB]/#MBC

[turned==true]

Turning

entry neighbours?n; turned = false

[n<alpha]

Turn180
entry move(0, av); wait (floor(180/av))

/turned = true

Ⓕ

ⓘ ●

RandomTurn
entry move(0, av); wait ([0,floor((360/av)/K)])

[n>=alpha]

/turned = true