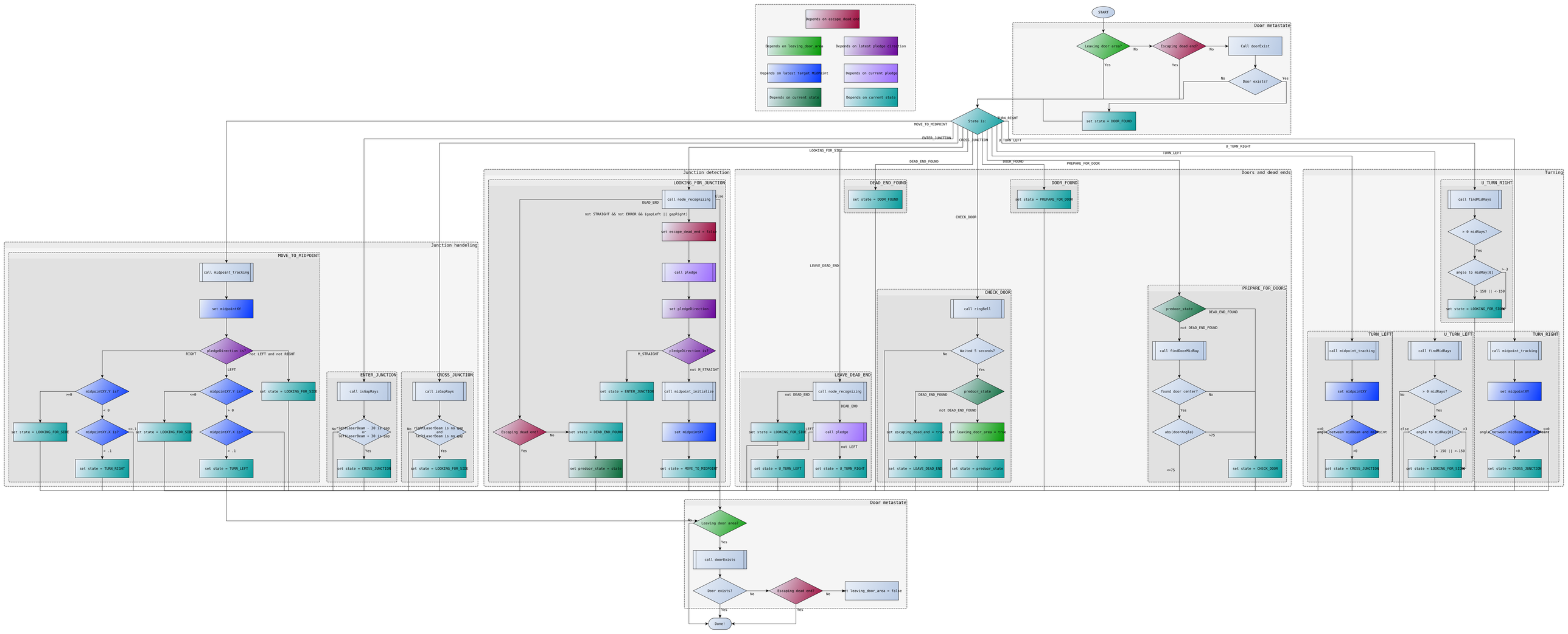
Depends on escape_dead_end
Depends on leaving_door_area
Depends on latest pledge direction
Depends on latest target Midpoint
Depends on current pledge
Depends on current state
Depends on current state
START
Door metastate
Leaving door area?
No
Escaping dead end?
No
Call doorExist
Yes
Yes
No
Door exists?
Yes
set state = DOOR_FOUND
State is:
MOVE_TO_MIDPOINT
TURN_RIGHT
ENTER_JUNCTION
CROSS_JUNCTION
U_TURN_LEFT
U_TURN_RIGHT
LOOKING_FOR_SIDE
TURN_LEFT
DEAD_END_FOUND
DOOR_FOUND
PREPARE_FOR_DOOR
Junction handeling
MOVE_TO_MIDPOINT
call midpoint_tracking
set midpointXY
pledgeDirection is?
RIGHT
LEFT
not LEFT and not RIGHT
set state = LOOKING_FOR_SIDE
midpointXY.Y is?
>=0
<=0
midpointXY.Y is?
< 0
> 0
set state = LOOKING_FOR_SIDE
set state = LOOKING_FOR_SIDE
midpointXY.X is?
>= .1
< -1
midpointXY.X is?
< .1
set state = TURN_RIGHT
set state = TURN_LEFT
ENTER_JUNCTION
call isGapRays
No
rightLaserBeam - 30 is gap
or
leftLaserBeam + 30 is gap
Yes
set state = CROSS_JUNCTION
CROSS_JUNCTION
call isGapRays
No
rightLaserBeam is no gap
and
leftLaserBeam is no gap
Yes
set state = LOOKING_FOR_SIDE
Junction detection
LOOKING_FOR_JUNCTION
call node_recognizing
else
DEAD_END
not STRAIGHT && not ERROR && (gapLeft || gapRight)
set escape_dead_end = false
call pledge
set pledgeDirection
pledgeDirection is?
M_STRAIGHT
not M_STRAIGHT
set state = ENTER_JUNCTION
call midpoint_initialize
set midpointXY
set state = MOVE_TO_MIDPOINT
Escaping dead end?
No
Yes
set state = DEAD_END_FOUND
set predoor_state = state
Doors and dead ends
DEAD_END_FOUND
set state = DOOR_FOUND
DOOR_FOUND
set state = PREPARE_FOR_DOOR
CHECK_DOOR
call ringBell
Waited 5 seconds?
No
Yes
predoor_state
DEAD_END_FOUND
not DEAD_END_FOUND
set escaping_dead_end = true
set leaving_door_area = true
set state = LEAVE_DEAD_END
set state = predoor_state
LEAVE_DEAD_END
call node_recognizing
not DEAD_END
DEAD_END
set state = LOOKING_FOR_SIDE
LEFT
call pledge
not LEFT
set state = U_TURN_LEFT
set state = U_TURN_RIGHT
PREPARE_FOR_DOORS
predoor_state
DEAD_END_FOUND
not DEAD_END_FOUND
call findDoorMidRay
found door center?
No
Yes
abs(doorAngle)
>75
<=75
set state = CHECK_DOOR
Turning
U_TURN_RIGHT
call findMidRays
> 0 midRays?
Yes
angle to midRay[0]
>-3
> 150 || <-150
set state = LOOKING_FOR_SIDE
TURN_LEFT
call midpoint_tracking
set midpointXY
>=0
angle between midBeam and Midpoint
<0
set state = CROSS_JUNCTION
U_TURN_LEFT
call findMidRays
> 0 midRays?
No
Yes
else
angle to midRay[0]
<3
> 150 || <-150
set state = LOOKING_FOR_SIDE
TURN_RIGHT
call midpoint_tracking
set midpointXY
>=0
angle between midBeam and Midpoint
<0
set state = CROSS_JUNCTION
Door metastate
No
Leaving door area?
Yes
call doorExists
Door exists?
No
Escaping dead end?
No
set leaving_door_area = false
Yes
Yes
Done!