VisualNav

× tgt: FireTarget, wm: WallMetrics, out: VNavOutput, enabled: boolean, centred: boolean
π CENTRE_TOL: real, YAW_TOL: real, TRACK_DIST: real, SCAN_DIST: real
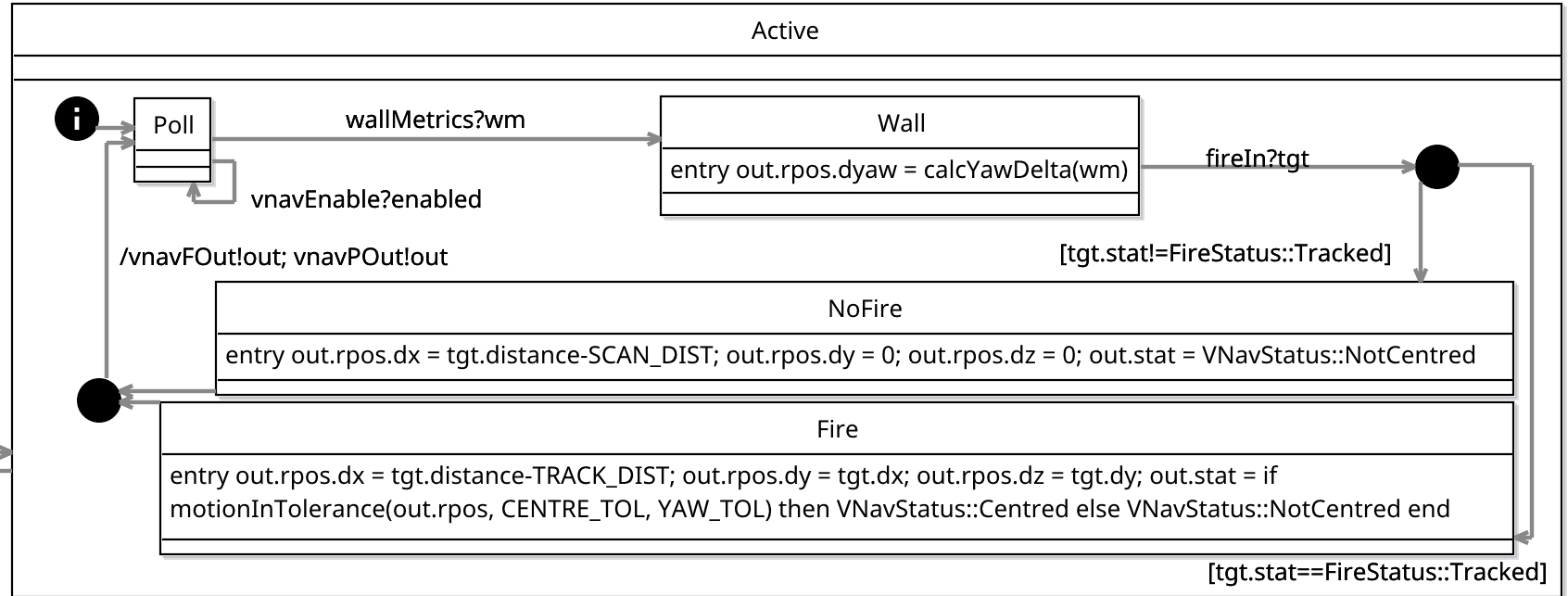ⓘ FireVisualEvents
ⓘ FireSubscriber
ⓘ VisualPlanningEvents

fireIn: FireTarget

vnavEnable: boolean

fireIn?tgt
vnavEnable?enabled
wallMetrics?wm

Inactive
entry out.stat = VNavStatus::Offline;
vnavFOut!out; vnavPOut!out

[enabled]

[not enabled]

Active

Poll

wallMetrics?wm

vnavEnable?enabled

/vnavFOut!out; vnavPOut!out

Wall
entry out.rpos.dyaw = calcYawDelta(wm)

fireIn?tgt

[tgt.stat!=FireStatus::Tracked]

NoFire
entry out.rpos.dx = tgt.distance-SCAN_DIST; out.rpos.dy = 0; out.rpos.dz = 0; out.stat = VNavStatus::NotCentred

Fire
entry out.rpos.dx = tgt.distance-TRACK_DIST; out.rpos.dy = tgt.dx; out.rpos.dz = tgt.dy; out.stat = if motionInTolerance(out.rpos, CENTRE_TOL, YAW_TOL) then VNavStatus::Centred else VNavStatus::NotCentred end

[tgt.stat==FireStatus::Tracked]

vnavFOut: VNavOutput

vnavPOut: VNavOutput

wallMetrics: WallMetrics

ƒₓ motionInTolerance(m: RelativeMotion, vec_tol: real, yaw_tol: real): boolean
◀ 0<=vec_tol/\0<=yaw_tol
▶ result==deltaInTolerance(m.dx, vec_tol)/\deltaInTolerance(m.dy, vec_tol)/\deltaInTolerance(m.dz, vec_tol)/\deltaInTolerance(m.dyaw, yaw_tol)

ƒₓ deltaInTolerance(d: real, tol: real): boolean
◀ 0<=tol
▶ result==(-tol<=d<tol)