Node

- parent: Node
- mid: Node
- left: Node
- right: Node
- type: Type

+ has_left(self) -> bool
+ has_mid(self) -> bool
+ has_right(self) -> bool
+ is_leaf(self) -> bool
+ is_root(self) -> bool

Tree

- root: Node
- current: Node
- T: map

+ append(self, _n: Node, _dir: DIRECTION) -> None
+ regress(self) -> None
+ set_current(self, node: Node) -> None
+ generate_node_id(self) -> str
+ DFSRec(self) -> None
+ build_tree_dict(self) -> map

Agent

+ controller: Controller
- logger: Logger

+ run(self) -> None
+ stop(self) -> None

Controller

- logger: Logger
- body: PhysicalBody
- tree: Tree
- store: Store

+ read_sensors(self) -> None
+ algorithm(self) -> bool
+ update_tree(self, actions, action_chosen) -> None
+ control_policy(self) -> tuple
+ decision_making_policy(self, com_actions) -> list
+ do_action(self, com_action) -> None
+ save_current_state(self) -> None
+ load_saved_state(self) -> None
+ rotate_to_final_g(self, vel, final_g) -> None
- __do_rotation(self, vel, c, degrees, final_g) -> None
- __rotate(self, vel, c: Clockwise, degrees) -> None
+ check_orientation(self, final_g, delta=2) -> None
+ adjust_orientation(self, final_g) -> None
+ compute_performed_degrees(c, init_g, curr_g) -> float
+ best_angle_and_rotation_way(self, init_g, final_g) -> float
+ goal_reached(self) -> bool
+ load_cfg_values(self) -> None

Logger

+ set_logfile(self, path: str) -> None
+ log(self, ...) -> None
+ is_loggable(self, ...) -> bool

PhysicalBody

- sim: SimConnection
- logger: Logger

+ virtual_destructor(self) -> None
+ move_forward(self, vel) -> None
+ move_backward(self, vel) -> None
+ turn(self, vel_rail_l, vel_rail_r) -> None
+ stop(self) -> None
- __set_motor_velocity(self, vel_FL, vel_FR, vel_RL, vel_RR) -> None
+ get_proxF(self) -> float | None
+ get_proxL(self) -> float | None
+ get_proxR(self) -> float | None
+ get_proxB(self) -> float | None
+ get_gate(self) -> bool
+ get_orientation(self) -> float
+ get_orientation_deg(self) -> float
- load_cfg_values()

Store

- state: State
- position: Position
- mode: Mode

+ store(self, state, position, mode) -> None
+ load(self) -> tuple

SimConnection

+ begin_connection(self) -> None
+ id(self) -> int
+ end_connection(self) -> None
+ start_simulation(self) -> None
+ pause_simulation(self) -> None
+ stop_simulation(self) -> None
+ get_object_handle(self, name: str) -> int | None

Usato da Controller

State (Enum)
+ STARTING: int
+ STOPPED: int
+ RUNNING: int
+ ROTATING: int
+ SENSING: int
+ BALANCING: int

Position (Enum)
+ UNKNOWN: int
+ INITIAL: int
+ CORRIDOR: int
+ JUNCTION: int

Mode (Enum)
+ EXPLORING: int
+ ESCAPING: int

Command (Enum)
+ START: int
+ STOP: int
+ RUN: int
+ ROTATE: int
+ GO_TO_JUNCTION: int
+ BALANCE: int

Compass (static class)
+ NORTH: int
+ SOUTH: int
+ EAST: int
+ WEST: int

+ string_to_compass(cmp: str)
+ compass_to_string(cmp)
+ compass_list_to_string_comma_sep(compass_list)
+ compass_list_to_concat_string(compass_list)

Clockwise (Enum)
+ LEFT: int
+ RIGHT: int