ClearpathBot + debris detector: DebrisDetector + debris remover: DebrisRemover + debris list : std::vector<string> + main(): int DebrisDetector - current image : cv::Mat - node options : rclcpp::NodeOptions - image subscriber_: image transport::Subscriber - rotate right : bool - rotate left : bool - move forward : bool - stop : bool - debris_detected_ : bool - current orientation : double - initial orientation : double - velocity_publisher_ : rclcpp::Publisher<geometry msgs::msg::Twist> - **debug image publisher** : rclcpp::Publisher<sensor msgs::msg::Image> - debris_detection_publisher_ : rclcpp::Publisher<std msgs::msg::Bool> - odometry subscriber : rclcpp::Subscription<nav msgs::msg::Odometry> + navigate to debris(): void + process image callback(image msg): void + process_odometry_callback(odom_msg) : void + detect and handle debris(): bool + move2next debris(): bool + get current image(): cv::Mat + get current orientation(): double + is_debris_detected(): bool + get rotate left(): bool + get_rotate_right(): bool + get stop(): bool + get move forward(): bool + set_velocity_publisher(publisher) : void

DebrisRemover

- scan : sensor msgs::msg::LaserScan

- unspawn client : rclcpp::Client<gazebo msgs::srv::DeleteEntity>

- timer : Timer

- remove debris node : rclcpp::Node::SharedPtr

- debris counter : int

- debris idx : std::list<int>

+ remove_debris(std::string) : bool

+ get recent debris(): int

+ get_debris_counter() : int

+ set debris counter(int count) : void

+ set unspawn client(rclcpp::Client<gazebo msgs::srv::DeleteEntity>

client): void

+ set_remove_debris_node(rclcpp::Node node) : void