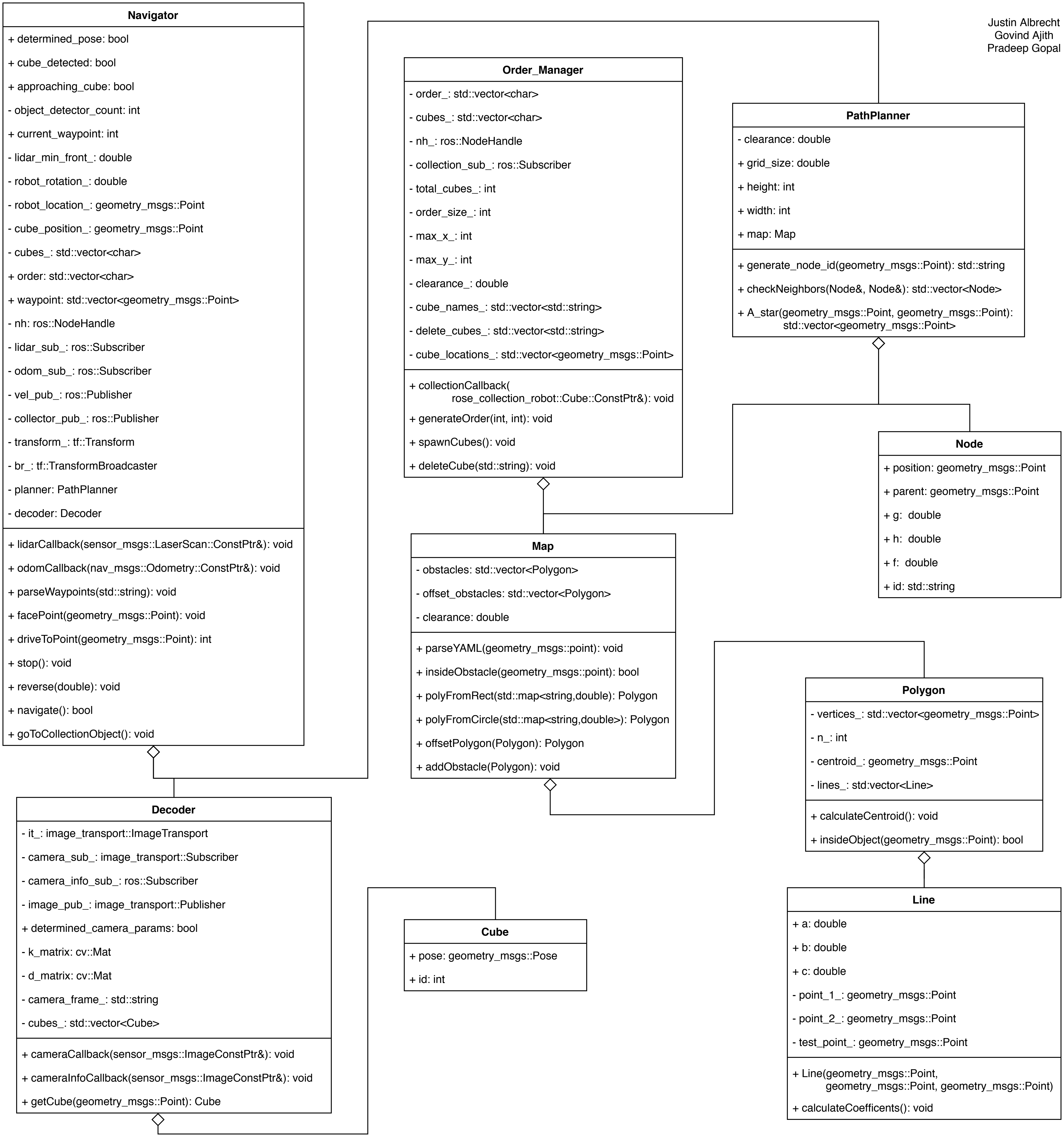
Justin Albrecht
Govind Ajith
Pradeep Gopal



Navigator

- + determined_pose: bool
- + cube_detected: bool
- + approaching_cube: bool
- - object_detector_count: int
- + current_waypoint: int
- - lidar_min_front_: double
- - robot_rotation_: double
- - robot_location_: geometry_msgs::Point
- - cube_position_: geometry_msgs::Point
- - cubes_: std::vector<char>
- + order: std::vector<char>
- + waypoint: std::vector<geometry_msgs::Point>
- - nh: ros::NodeHandle
- - lidar_sub_: ros::Subscriber
- - odom_sub_: ros::Subscriber
- - vel_pub_: ros::Publisher
- - collector_pub_: ros::Publisher
- - transform_: tf::Transform
- - br_: tf::TransformBroadcaster
- - planner: PathPlanner
- - decoder: Decoder

- + lidarCallback(sensor_msgs::LaserScan::ConstPtr&): void
- + odomCallback(nav_msgs::Odometry::ConstPtr&): void
- + parseWaypoints(std::string): void
- + facePoint(geometry_msgs::Point): void
- + driveToPoint(geometry_msgs::Point): int
- + stop(): void
- + reverse(double): void
- + navigate(): bool
- + goToCollectionObject(): void

Order_Manager

- - order_: std::vector<char>
- - cubes_: std::vector<char>
- - nh_: ros::NodeHandle
- - collection_sub_: ros::Subscriber
- - total_cubes_: int
- - order_size_: int
- - max_x_: int
- - max_y_: int
- - clearance_: double
- - cube_names_: std::vector<std::string>
- - delete_cubes_: std::vector<std::string>
- - cube_locations_: std::vector<geometry_msgs::Point>

- + collectionCallback(rose_collection_robot::Cube::ConstPtr&): void
- + generateOrder(int, int): void
- + spawnCubes(): void
- + deleteCube(std::string): void

PathPlanner

- - clearance: double
- + grid_size: double
- + height: int
- + width: int
- + map: Map

- + generate_node_id(geometry_msgs::Point): std::string
- + checkNeighbors(Node&, Node&): std::vector<Node>
- + A_star(geometry_msgs::Point, geometry_msgs::Point): std::vector<geometry_msgs::Point>

Node

- + position: geometry_msgs::Point
- + parent: geometry_msgs::Point
- + g: double
- + h: double
- + f: double
- + id: std::string

Map

- - obstacles: std::vector<Polygon>
- - offset_obstacles: std::vector<Polygon>
- - clearance: double

- + parseYAML(geometry_msgs::point): void
- + insideObstacle(geometry_msgs::point): bool
- + polyFromRect(std::map<string,double): Polygon
- + polyFromCircle(std::map<string,double>): Polygon
- + offsetPolygon(Polygon): Polygon
- + addObstacle(Polygon): void

Polygon

- - vertices_: std::vector<geometry_msgs::Point>
- - n_: int
- - centroid_: geometry_msgs::Point
- - lines_: std:vector<Line>

- + calculateCentroid(): void
- + insideObject(geometry_msgs::Point): bool

Decoder

- - it_: image_transport::ImageTransport
- - camera_sub_: image_transport::Subscriber
- - camera_info_sub_: ros::Subscriber
- - image_pub_: image_transport::Publisher
- + determined_camera_params: bool
- - k_matrix: cv::Mat
- - d_matrix: cv::Mat
- - camera_frame_: std::string
- - cubes_: std::vector<Cube>

- + cameraCallback(sensor_msgs::ImageConstPtr&): void
- + cameraInfoCallback(sensor_msgs::ImageConstPtr&): void
- + getCube(geometry_msgs::Point): Cube

Cube

- + pose: geometry_msgs::Pose
- + id: int

Line

- + a: double
- + b: double
- + c: double
- - point_1_: geometry_msgs::Point
- - point_2_: geometry_msgs::Point
- - test_point_: geometry_msgs::Point

- + Line(geometry_msgs::Point, geometry_msgs::Point, geometry_msgs::Point)
- + calculateCoefficents(): void