Justin Albrecht
Govind Ajith
Pradeep Gopal

Map

```
+ obstacles: std::vector<Polygon>
+ offset_obstacles: std::vector<Polygon>
+ clearance: double
---
+ inside_obstacle(geometry_msgs::point): bool
- poly_from_rect(std::map<string,double): Polygon
- poly_from_circle(std::map<string,double>): Polygon
- circle_vertices(geometry_msgs::point): bool
- offset_polygon(Polygon): Polygon
```

PathPlanner

```
+ map: Map
---
+ A_star(geometry_msgs::Point, geometry_msgs::Point): std::vector<geometry_msgs::Point>
+ euler_path(): std::vector<geometry_msgs::Point>
```

Polygon

```
+ vertices: std::vector<geometry_msgs::Point>
+ n: int
+ centroid: geometry_msgs::Point
+ lines: std:vector<Line>
---
+ calculate_centroid(): void
+ inside(geometry_msgs::Point): bool
```

Line

```
+ end: geometry_msgs::Point
+ A: double
+ B: double
+ C: double
---
+ Line(geometry_msgs::Point, geometry_msgs::Point, geometry_msgs::Point): void
+ coefficients(): void
```

Navigator

```
+ found_object: bool
+ path_planner: PathPlanner
+ euler_waypoints:  std::vector<geometry_msgs::Point>
+ current_waypoint:  int
+ lidar_sub: ros::Subscriber
+ odom_sub: ros::Subscriber
+ transform: tf::Transform
+ br: tf::TransformBroadcaster
+ nh: ros::NodeHandle
---
+ lidar_callback(sensor_msgs::LaserScan &msg): void
+ check_for_collection_object(std::vector<double>): void
+ follow_euler_path(): void
+ go_to_collection_object(): void
+ drive_to_drop_off(): void
+ return_to_euler_path(): void
```

Controller

```
+ cmd_vel_pub: ros::Publisher
---
+ drive_to_waypoint(geometry_msgs::Pose): void
```

Manipulator

```
+ pick_waypoints: std::vector<geometry_msgs::Pose>
+ place_waypoints: std::vector<geometry_msgs::Pose>
+ end_effector_pub: ros::Publisher
---
+ pick_part(): void
+ place_part(): void
```

Decoder

```
+ camera_sub: ros::Subscriber
+ order_sub: ros::Subscriber
---
+ camera_callback(sensor_msgs::Image &msg): void
+ decode_qr(sensor_msgs::Image): std::string
+ in_order(std::string): bool
```

Order_Manager

```
+ order: std::vector<int>
+ order_pub: ros::Publisher
+ nh: ros::NodeHandle
---
+ generate_order(int): void
+ spawn_cubes(): void
```