TerpRescue - tag : struct - tagList : vector<tag> - robotPose : geometry_msgs::Pose - rawMap : nav_msgs::OccupancyGrid - tagMarkers : visualization msgs::MarkerArray - lidar : vector<float> - markerList : std::vector<ar track alvar msgs::AlvarMarker> - tagLocalizer : Localizer - botOdom: nav msgs::Odometry - robotVelocity: geometry msgs::Twist - tagLocalizer : Localizer - explorer : Explorer - lidarSubscriber : ros::Subscriber - arSubscriber : ros::Subscriber - odomSubscriber : ros::Subscriber - mapSubscriber : ros::Subscriber - tagPublisher : ros::Publisher - vel pub : ros::Publisher - defaultLinearSpeed : double - defaultAngularSpeed : double + TerpRescue() + getPointDistance(geometry_msgs::Point, geometry_msgs::Point) : double + visualization(): void + detectTags(): void + getLidar(): vector<float> + getCamera(): sensor_msgs::Image + getRawMap(): nav_msgs::OccupancyGrid + getRobotPose(): geometry_msgs::Pose + getTagList(): vector<struct> + getMarkerList(): std::vector<ar_track_alvar_msgs::AlvarMarker> + getTagWorldTransformList(): std::vector<tf2::Transform> - lidarCallback(sensor msgs::LaserScan, this): void - arPoseCallback(sensor msgs::Image, this): void - botOdomCallback(nav_msgs::Odometry, this): void - mapCallback(nav msgs::OccupancyGrid,this): void

Localizer + recognizeTag(std::vector<ar_track_alvar_msgs::AlvarMarker>): bool + locateTag(std::vector<ar_track_alvar_msgs::AlvarMarker>): std::vector<tf2::Transform> + transformationTagPosition(const std::vector<ar_track_alvar_msgs::AlvarMarker>&, const nav_msgs::Odometry): std::vector<tf2::Transform>

