## TerpRescue - nh : ros:NodeHandle - tag : struct - tagList : std::vector<tag> - robotPose : geometry msgs::Pose - rawMap: nav msgs::OccupancyGrid - synthesizedMap: nav msgs::OccupancyGrid - tagMarkers : visualization msgs::MarkerArray - lidar : std::vector<float> - markerList : std::vector<ar track alvar msgs::AlvarMarker> - botOdom: nav\_msgs::Odometry - robotVelocity : geometry\_msgs::Twist - defaultLinearSpeed : double - defaultAngularSpeed : double tagLocalizer : Localizer - explorer : Explorer - lidarSubscriber : ros::Subscriber - arSubscriber : ros::Subscriber - odomSubscriber : ros::Subscriber - mapSubscriber : ros::Subscriber - tagPublisher : ros::Publisher - vel\_pub : ros::Publisher - lidarCallback(sensor msgs::LaserScan, this): void - arPoseCallback(sensor\_msgs::Image, this): void - botOdomCallback(nav\_msgs::Odometry, this): void - mapCallback(nav\_msgs::OccupancyGrid,this): void + TerpRescue() + getPointDistance(geometry msgs::Point, geometry msgs::Point) : double + visualization(): void + detectTags(): void + getTagList(): std:vector<struct> + getMarkerList(): std::vector<ar\_track\_alvar\_msgs::AlvarMarker> + getTagWorldTransformList(): std::vector<tf2::Transform>

## 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<ff2::Transform>

