Geofencing application for mobile robots

Case 1

The geofenced area is defined in the odometry frame of the robot i.e. the initial pose of the robot is the origo of the coordinate frame and the limits of the geofencing polygon are defined w.r.t. this frame. Subsequently, the geofenced area is *not* fixed in global coordinates, but depends on the initial pose of the robot.

Example

The robot may not move further than 5m from it's initial starting position.

Requirements

- define geofencing limits when launching the node
- subscribe to velocity command messages
- check whether robot is inside the geofenced area
- if the geofenced area is breached, make necessary changes to velocity commands
- publish velocity command messages to robot
- visualize the geofenced area in RViZ

Case 2

The geofenced area is defined in the map frame, i.e. it *is* fixed in global coordinates. In order to respect the geofencing boundaries, the robot needs to localize itself in the map frame. The map and initial pose of the robot are known.

Example

The robot must stay within the Konetalo lobby and may not enter any of the corridors or come within 2m of the stairway.

Additional requirements

- robot localization
- possibility to define geofencing frame when launching the node

Existing ROS packages, such as amcl, may be used for the robot localization. Check e.g. `turtlebot_navigation` tutorial.

Additional tasks according to time and motivation

- geofenced area may be defined by clicking desired points in RViZ
- own implementation of robot localization