**Notes**

At some point, the user has to assign ROS topics to specific things on the robot so the plugins know what to listen on. Maybe assign defaults at robot design time and allow changing during runtime?

Plugins should report how many channels are needed and what they are used for

Maybe allow optional channels? (Like reporting simulated motor current)

Actual wheel velocities

**General Simulation Loop**

Control code publishes target motor velocities

Plugins turn target motor velocities into robot velocities

Physics engine ticks; sets robot positions, sets actual robot velocities

Plugins take robot + map info; publish response to control code and generate OpenGL display objs

UI draws robots + extra objects from plugins

Actual velocity

Target wheel velocities

Lidar info

Sensor Location + map info

Target velocity

Visual of lidar rays

Can be built into robots

Main Screen

Start Simulator

Open robot editor

Open map importer/editor

Example Lidar plugin

Given lidar position and angle, plus map -> sends ranging info to control code & generates on-screen rays

Example Differential Drive plugin

Listens for velocity set messages from control code

Provides robot xdot, ydot, thetadot values

UI

OpenGL with QML overlay buttons

Load/import map

Load/import robot design

Add robots from imported list to simulation

Display robots on map

Optionally display sensor data

Physics Engine

Keep track of absolute position and rotation of robots, sensors, and map objects

Move robots (and object?) based on robot shape and robot target velocity x, y, theta

Simulation

Map Importer

Import map from rasterized image

Save map as custom format for easy re-loading?

Allow editing of map to add things?

Export map to current simulation

Robot Editor

Load/Save to file

Allow placement/selection of drive train

Allow placement/selection of sensors

Maybe just defined primitives for robot shape?

Export to current simulation

Main Application