BOT-DRIVES-TO-WAYPOINT

A custom wheeled robot in Gazebo's postoffice world. When the DRIVE button is pressed, the robot detects the RED gate, and drives toward it. Other functionality helps it get to its goal.

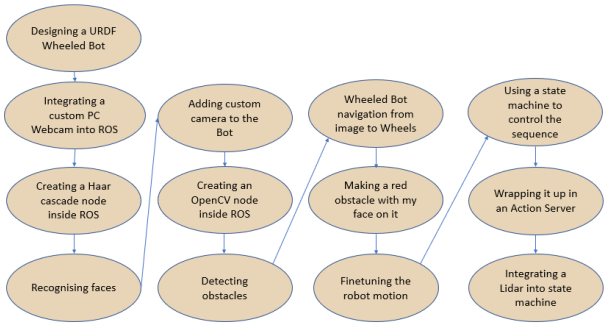
Github : BotDrivestoWaypoint

https://github.com/AlphaPilotFrance/BotDrivestoWaypoint

A custom wheeled robot in Gazebo's postoffice world. When the DRIVE button is pressed, the robot detects the RED gate, and drives toward it. Other functionality helps it get to its goal.

README INCLUDES :

- Demo
- Project Stack
- Current Functionality
- Contributions are welcome
- Official Changelog
- The Steps to Usability



DESIGNING A URDF WHEELED BOT

I used the ROS manual for this, which you can find in our ROS Resources Page.

- FIND EXACT PAGES

You can easily get up and going with the finished version of the URDF accessible on our github or, if needed, from the authors themselves.

- FIND GITHUB

A more in-detail explanation of this process can be found among our tutorials here.

- LINK TO OUR TUTORIAL

INTEGRATING A CUSTOM PC WEBCAM INTO ROS

I used a ROS node. Instructions for use can be found here.

- FIND NAME OF ROS NODE.

It's possible to visualise the output using RVIZ.

- VISUALIZE OUTPUT.

CREATING A HAAR CASCADE NODE INSIDE ROS

I replicated a Haar cascade from this project.

- FIND HOW I SET IT UP.

Haar cascades have a particular architecture which would be useful to learn about, here.

- ADD PICTURE AND EXPLANATION

RECOGNISING FACES

A little video shows the car approaching the face.

- ADD THE VIDEO

For recognising any other object (hand gestures, etc), refer to this series of videos.

- REFER TO VIDEO.

ADDING CUSTOM CAMERA TO THE BOT

I used a Gazebo tutorial in order to add a sensor.

- GAZEBO TUTORIAL

The final camera gives images accurately.

- RVIZ OF CAMERA

CREATING AN OPENCV NODE INSIDE ROS

There was no support for OpenCV6 with Python2, therefore a ROS node using OpenCV6 was put together using various resources.

- DIRECT LINK TO THE NODE.
- WHAT PROJECTS I TOOK FROM.

DETECTING OBSTACLES

For detecting obstacles, I used some basic OpenCV.

- RVIZ OUTPUT
- OPENCV SCRIPT ANNOTATED

MAKING A RED GATE WITH MY FACE ON IT

For creating an obstacle, I learned how to build a Rigid Body in Gazebo.

- TUTORIAL TO CREATE BODY AND TO ADD A PICTURE
- MY FINAL PRODUCT

WHEELED BOT NAVIGATION FROM IMAGE TO WHEELS

Different behaviours had to be coded according to what was detected in the images.

- MY ANNOTATED CODE
- THE RESULT

FINETUNING THE ROBOT MOTION

The robot motion required different speeds for the different parts of the trajectory. Only with specific tuning could I achieve a bot driving toward the gate.

- MY ANNOTATED CODE
- WHAT A CHANGE LOOKS LIKE

USING A STATE MACHINE TO CONTROL THE SEQUENCE

Splitting the robot motion into states made it possible to link ROS data to a state machine.

- MY ANNOTATED CODE
- LINK TO THE TURTLESIM TUTORIALS

WRAPPING IT UP IN AN ACTION SERVER

The state machine could be attached to an action server.

- THE ANNOTATED CODE
- HOW THIS ENABLES CODE EXECUTION

INTEGRATING A LIDAR INTO STATE MACHINE

Using this ROS stack, we can now add more modules into our code.

- PLACES WHERE THE LIDAR IS INTEGRATED
- FURTHER INFO ON LIDARS