*MARCO-POLO GAME FOR ROBOT

YUXING LIU & AIDAN GOROWAY

DESCRIPTION 1

This project aims to navigate the robot by verbal commands:

- People will give instruction commands, similar to “Robot, please go forward”, “Robot, please turn left”, “Robot, please turn right 30 degrees and go forward 2 meters”…
- The speech recognition will convert the verbal commands into text
- The robot will act on the text to navigate
- The end goal for the robot is to be able to navigate to any goal at an unspecified location, using only the vocal directions we give it

DESCRIPTION 2

To use this project:

- install DeepSpeech Package
- run launchfile(not working, see README)
- give verbal commands through microphone
  - command format:
    - Turning direction(right/left) + degrees(30°, 60°, or 90°)
      - eg. Robot, please turn right 30°
      - if no specific degrees are given, the robot will turn 90°
    - Movement direction(forward, backward) + distance(1-5 meters)
      - eg. Robot, please go forward 4 meters
      - if no specific meters are given, the robot will go 1 meter
    - Turning direction + degrees + Movement direction + distance
      - eg. Robot, please turn right 30° and go forward 4 meters
      - eg. Robot, please turn right and go backward

SPEECH RECOGNITION

The robot uses DeepSpeech in order to convert human speech into text. We then take that text and get arguments for navigation from it.

- The process goes as follows (All of the following fall within audiorec.py):
  - User talks into microphone ->
  - Audio is saved as a .wav file ->
  - DeepSpeech takes .wav file and gets text from it ->
  - Text is parsed for specific keywords ->
  - Keywords are provided as parameters for navigation functions (navigationRev1.py)

STRING MANIPULATION BASICS

In stringManip.py, the following functions exist:

- The grabParams() function tries to grab the parameters(direction, degree, isForward, distance) of robot navigation from the text.
- rotationCheck() and moveCheck() are helper functions that check to see if the user specified a degree of rotation and a movement distance
  - degreeHelper() and distanceHelper() check that specification to see if its valid
- The speaker() function publishes the String commands of the navigation parameters to the speakCommand topic

STRING MANIPULATION SPECIFICS

There are a few important details to note about the specifics of how the raw audio is converted into parameters used to navigate with:

- In ambiguous cases of “plurals” (forward vs forward**s**), (degree vs degree**s**), (meter vs meter**s**), we check for use of both pronunciations.
- Deepspeech interprets numbers with their English spelling (four, thirty, thirty two, ect), so we limited the degree count to 30, 60, or 90. We manually convert these spellings to string-ified integers (1, 3, 60, ect)
    - Similarly, meters can only be provided with a number 1-5. We do the same thing here.

NAVIGATION

Navigating the robot:

- The moveRobot function in navigation.py subscirbe the speakCommand topic, get the String commands, and publish movement to cmd_vel
- The navigation function is given 4 navigation parameters:
  - Turning direction (left/right/no):
    - Degrees of rotation (30°, 60°, or 90°)
  - Movement direction (forward/backward/no):
    - Distance (1-5 meters)
- The robot also gets the minimum distance to any obstacle within a 30° cone from its front and back (from the Lidar sensor), in order to detect and avoid colliding with an obstacle.
  - Specifically, if an obstacle is within 0.5 meters of the robot's movement path, robot will stop, and also print "There is an obstacle!!!"

VIDEOS

Work in simulation:
https://drive.google.com/file/d/16TNGaBfkRSDH0U13EaK5qjOrtH3j8n_N/view?usp=drive_link

Work in real word:
https://drive.google.com/file/d/1iaDmU14ihQVJmPui6By-aMIJAmOQkksa/view?usp=drive_link

WHAT DIDN’T WORK

There are a few things that didn’t work out that should be mentioned

- The first thing to mention is that the original idea for this project was a modified game of Marco-Polo, but that changed.
  - This is first because the microphones do not actually serve as sound sensors.
  - Second, it was infeasible to mount the microphone on the robot, so calling out commands to the robot was scrapped as a concept.
- The second big thing to mention is that while we DO have a launchfile, it does not function. This is not ideal, we know.
- While using pyAudio, we encountered several interesting bugs, which you might know of.
  - One makes it so the .wav file recorded is fast and high-pitched, like chipmunk-speak.
  - Another makes it so the file just... stops.
  - Having an easy time recording the audio and a nightmarish time recording the audio is a coinflip sometimes.
- There are some other non-bug eccentricities with DeepSpeech to point out as well, like how it tends to hear “two” as “to” more often than not. “Four” sometimes became “for”. One time, “four meters” was picked up as “formulas”