CompRobo Final Project Story 1

SeungU Lyu, HK Rho

Using April Tag to localize the robot

1. Decided to use April Tag for the localization method.
2. Went through camera calibration to get the camera parameters of the camera attached to the Neatos.
3. Retrieved transformation matrix between the camera frame and the April tag. Used Python bindings for Apriltag3 library to achieve this.

Project Planning - Driving, Recording

1. Took ideation sessions together to figure out what is needed for the project. Decided that we will need map recording, and driving based on the already saved map.
2. Plan for record map
3. Plan for drive neato
4. Decided to use transformation matrix to create map frame. We had hard time understanding how transformation matrix work, and how to implement them. By going to Paul's office hours, we were able to get basic understanding to start the implementation. Plan for record map, plan for driving, how to convert from tag frame to base frame

Driving and Recording Implementation / Environment Setup

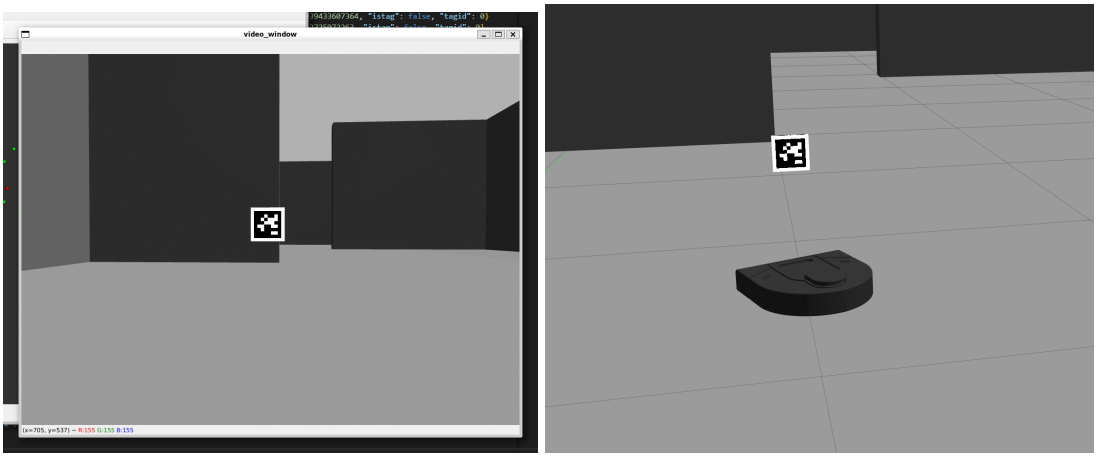
1. Started with implementing record map. Decided that the map with be saved as a JSON file with list of points in the map frame. When the user clicks the OpenCV window, recording starts and the robot's current position is set as the map frame's origin. Every

tag/point is saved based on this origin.

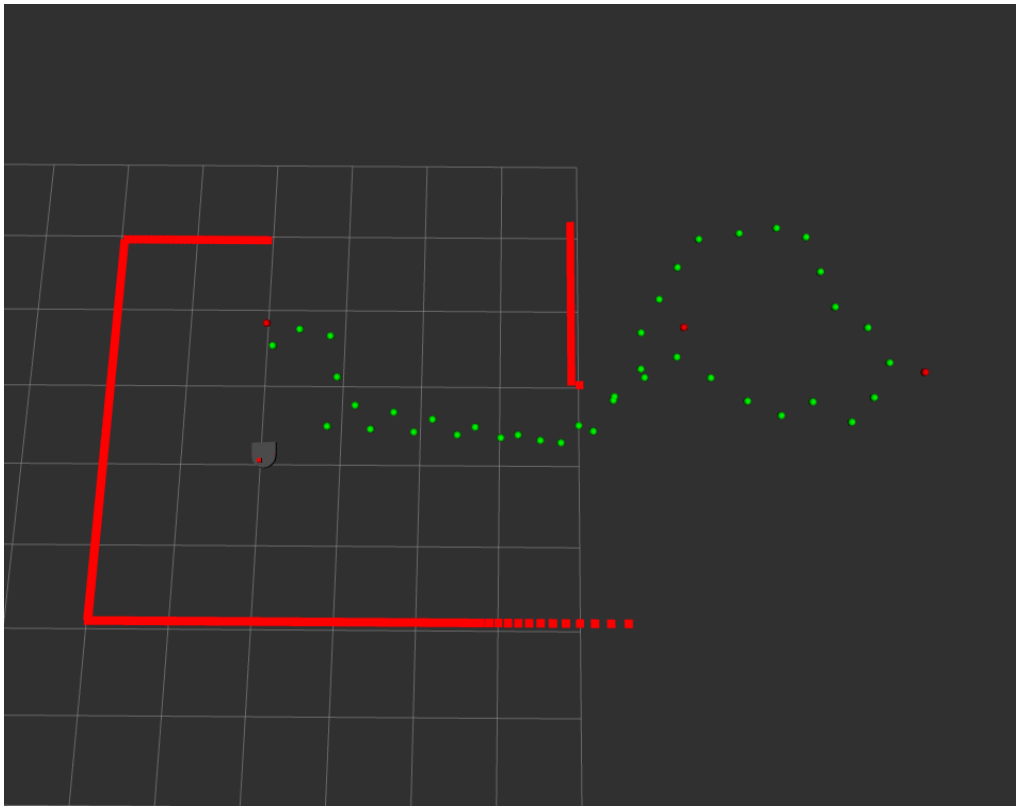
ros2_ws > src > NeatoKart > maps > {} test5.json > ...


```
{"x": 0.548007471346073, "y": 0.0001320384190584000 8, "theta": 8.927556742244066e-05, "istag": false, "tagid": 0}
{"x": 1.0500806105892373, "y": 0.0005092389001342124, "theta": 0.0013533046760987268, "istag": false, "tagid": 0}
{"x": 1.550138222984926, "y": 0.0011847904066235593, "theta": 0.001348932246354016, "istag": false, "tagid": 0}
{"x": 2.100125437443626, "y": 0.0019260122644118027, "theta": 0.001343923889482128, "istag": false, "tagid": 0}
{"x": 2.437304352559016, "y": -0.0004568848619308507 6, "theta": -2.012011514280496, "istag": false, "tagid": 0}
{"x": 2.119852924943732, "y": -0.3995379456760366, "theta": -2.264367823408594, "istag": false, "tagid": 0}
{"x": 1.8001961617504314, "y": -0.784051468389332, "theta": -2.2643727896095323, "istag": false, "tagid": 0}
{"x": 1.4485942393789313, "y": -1.2069724760636027, "theta": -2.2643765820843615, "istag": false, "tagid": 0}
{"x": 1.0969895420562996, "y": -1.6298887770731794, "theta": -2.2643828902267784, "istag": false, "tagid": 0}
{"x": 0.7453818156841383, "y": -2.052806122674141, "theta": -2.2643851197529323, "istag": false, "tagid": 0}
{"x": 0.393774157882528, "y": -2.475721845283169, "theta": -2.2643869104124836, "istag": false, "tagid": 0}
{"x": 0.042165645438370314, "y": -2.8986371277890184, "theta": -2.2643887249562185, "istag": false, "tagid": 0}
{"x": -0.3094435762563892, "y": -3.321551671911566, "theta": -2.2643914232066047, "istag": false, "tagid": 0}
{"x": -0.6610528307158191, "y": -3.7444637199439077, "theta": -2.2643918739590627, "istag": false, "tagid": 0}
{"x": -1.0126647737012222, "y": -4.167378773243394, "theta": -2.2643951223283865, "istag": false, "tagid": 0}
{"x": -1.3642759979235575, "y": -4.590290645940495, "theta": -2.264397995452593, "istag": false, "tagid": 0}
{"x": -1.715890205182015, "y": -5.013201215200811, "theta": -2.264403219334634, "istag": false, "tagid": 0}
{"x": -1.9977100642656225, "y": -4.779928703614702, "theta": 2.1243924418746976, "istag": false, "tagid": 0}
{"x": -2.053749945794487, "y": -4.287381301098614, "theta": 2.129088379213591, "istag": false, "tagid": 0}
{"x": -2.0394853451761072, "y": -3.780484411202621, "theta": 2.1041191383440254, "istag": false, "tagid": 0}
{"x": -2.045571720054362, "y": -3.251198116489861, "theta": 1.8349175333328092, "istag": false, "tagid": 0}
{"x": -2.0301131463428552, "y": -2.725343084960661, "theta": 1.8290494647088085, "istag": false, "tagid": 0}
{"x": -2.014651257474445, "y": -2.1985635558177545, "theta": 1.8230258589571793, "istag": false, "tagid": 0}
{"x": -1.9991837175739264, "y": -1.670858319378219, "theta": 1.8168703227298122, "istag": false, "tagid": 0}
{"x": -1.9837128156590325, "y": -1.1422221640354842, "theta": 1.8105670913567018, "istag": false, "tagid": 0}
{"x": -1.9682394765634172, "y": -0.6126586215746497, "theta": 1.8041117231699404, "istag": false, "tagid": 0}
{"x": -1.952764090599696, "y": -0.08216288219399415, "theta": 1.7975065383249433, "istag": false, "tagid": 0}
{"x": -1.93728907045857, "y": 0.4492611104817694, "theta": 1.7907356478225755, "istag": false, "tagid": 0}
{"x": -1.9218195224232175, "y": 0.9816169234444604, "theta": 1.7837601644800085, "istag": false, "tagid": 0}
{"x": -1.9063682103273694, "y": 1.5149276464642938, "theta": 1.7764759266219217, "istag": false, "tagid": 0}
{"x": 1.3198708877622751, "y": -0.03429509129640951, "theta": -0.025075089200655588, "istag": true, "tagid": 0}
```

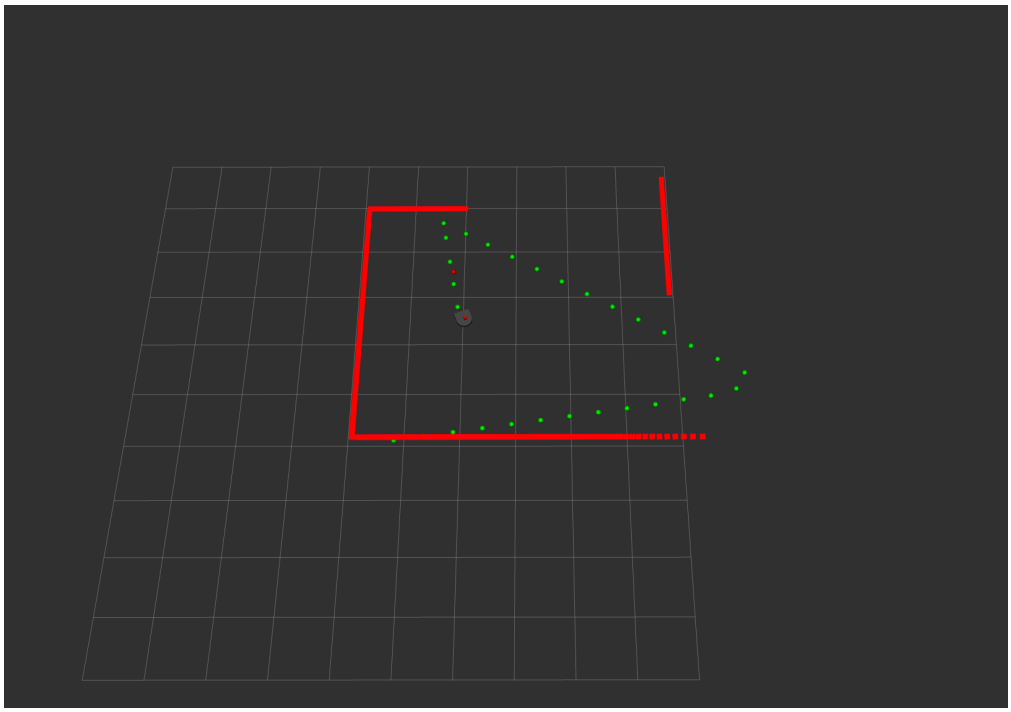

2. Implemented driving neato, which will going to read the saved map data and will find the odom position of all the points based on tag's pose.
3. For quick testing purpose, [forked and changed neato_packages repo] and added april tags to the maze world.



4. Tested the code, fixed some bugs, and it seems like the map is loading correctly and the robot can rocalize itself.



map file



map file

5. The biggest issue right now is that the simulation environment does not have correct camera calibration, and sometimes the angle value we get from april tag is strange. Also,

there are bug where the map flips whenever april tag returns angle that is more than 180 degrees, which we are trying to fix right now.

Track & Item Generation

1. **Track Generation**
   Neato is given a set of points that forms the track at every frame. Given this information, we need to re-draw the track at every updated Neato position. For drawing the track based on the Neato camera's view, we will utilize the approach used in a past CompRobo project.
2. **Item Generation**
   ** This is a stretch goal **
   If the track contains items/obstacles, they will be shown in the Neato camera. To do so, we will utilize the API or the approach from another past CompRobo project.