Indoor Localization Android App

Zixin Yin 10/15/2021

- Finished translating all code to java
- Pattern worked
- readTDoANew; readInitPower tested

- To Do
- Test antennaCorrection
- fsolve

Next paper(extend the app)

- Path planning done offline on the mobile phone
- How to put context into the map make it perceivable by tag
- System learns what each room does itself(learn by twitters and maps)(evolve on its own)(collaborate with calendar meeting time and location to learn the map)
- Learn about how google map know semantic of places
- How google map get 3D map(vehicle with 360 degree camera, tag semantics based on the view)

- Solve tag location using 3 sensors: (9) r1—> (10)
- Problem: eliminate r1 reading *A Passive Localization Algorithm and Its Accuracy Analysis —> does not solve for (x,y)*
- *Cannot solve r1*
- *Looking for other solutions for hyperbolic system*
- *Continue with 3 sensors: solve for r1 first; test*

Ionic

- Installed node js; cordova (npm install -g cordova)
- Install script for nvm: curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.0/install.sh | bash
- Load nvm: export NVM_DIR="$([-z "${XDG_CONFIG_HOME-}"] && printf %s "${HOME}/.nvm" || printf %s "${XDG_CONFIG_HOME}/nvm")"
- [-s "$NVM_DIR/nvm.sh"] && \. "$NVM_DIR/nvm.sh" # This loads nvm
- Install nvm: curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash
- Check if nvm installed properly: command -v nvm (if something prints, installed properly)
- To download and install the latest LTS release of NodeJS: nvm install —lts
- install ionic with npm: npm install -g @ionic/cli
- Start a project: cd Desktop
- Ionic start —list (show all options for building apps)

  Ionic start maps —type=angular

  Cd maps

  Ionic serve

  On webpage: 3dots—>more tools—>developer tools

  Terminal: ctrl c

  Ionic serve -l (start ionic with lab which makes the app look more like a mobile app)

Ionic in visual studio

- Ionic cordova platform add iOS (add platform for iOS)(*ionic cordova platform rm iOS* will remove platform)
- Ionic cordova platform add browser (add browser as a platform)
- Ionic serve (run the app) ; ionic serve -l (run the app in ionic lab)

Generate a page

Ionic g
Error: cannot generate page
Solved using: npm i @ionic/angular-toolkit@2.3.0 —save-dev

Problem

ionic cordova emulate iOS:
[ERROR] **An error occurred while running subprocess ng.**

ng run app:ionic-cordova-build --platform=ios exited with exit code 127.

Re-running this command with the --verbose flag may provide more information.

Solution use ionic serve

Installed ionic and nvm but command not found for every new command window
Solution: add command *export NVM_DIR="$HOME/.nvm"*
[-s "$NVM_DIR/nvm.sh"] && \. "$NVM_DIR/nvm.sh" to ~/.zshrc file (edit file from terminal using *vim ~/.zshrc*

App Development Idea

- *Frontend* —> Run in mobile devices;

  Send inputs, e.g. signal strength, anchor locations, to backend server

  Built in Ionic with html, css, javascript

- *Backend* —> Run in server device (can use laptop for experiments)

  Receive input from mobile device, process data, calculate tag location, and send result back to frontend

  Written in Python

Advantages: separating frontend from backend removes workload from cellphones and avoids huge battery cost on mobile device;

Backend Code (Reader)

- Read anchor and signal strength information from file successfully; checked length and content of output

```
30
31      print(len(read_packet_id))
32      print(len(read_ID1))
33      print(len(read_ID2))
34      print(len(read_DDoA))
35      print(len(read_FP_PW_tag))
36
37      print(len(init_packet_id))
38      print(len(init_FP_PW))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
yinzixin@lawn-143-215-90-164 indoorbackend % python read.py
16052
16052
16052
16052
16052
4782
4782
yinzixin@lawn-143-215-90-164 indoorbackend %
```

Backend Code (anchor correction)

- Anchor correction with antenna delays gives proper result

```
    print corrected_ddoa

antenna_delays = [-514.800046735725,
-515.807752377787,
-515.311106592656,
-515.115189872074,
-514.882780169421,
-513.592856014242,
-514.918087972098,
-515.128385013776,
-514.793541561308,
-515.321036364222,
-515.241712743109,
-514.973272610434,
-514.989929612259]
antenna_correct_ddoa(556,2,3,antenna_delays)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
yinzixin@lawn-143-215-90-164 indoorbackend % python correction.
29.3671776133
yinzixin@lawn-143-215-90-164 indoorbackend %
```

Backend Code (tag solver)

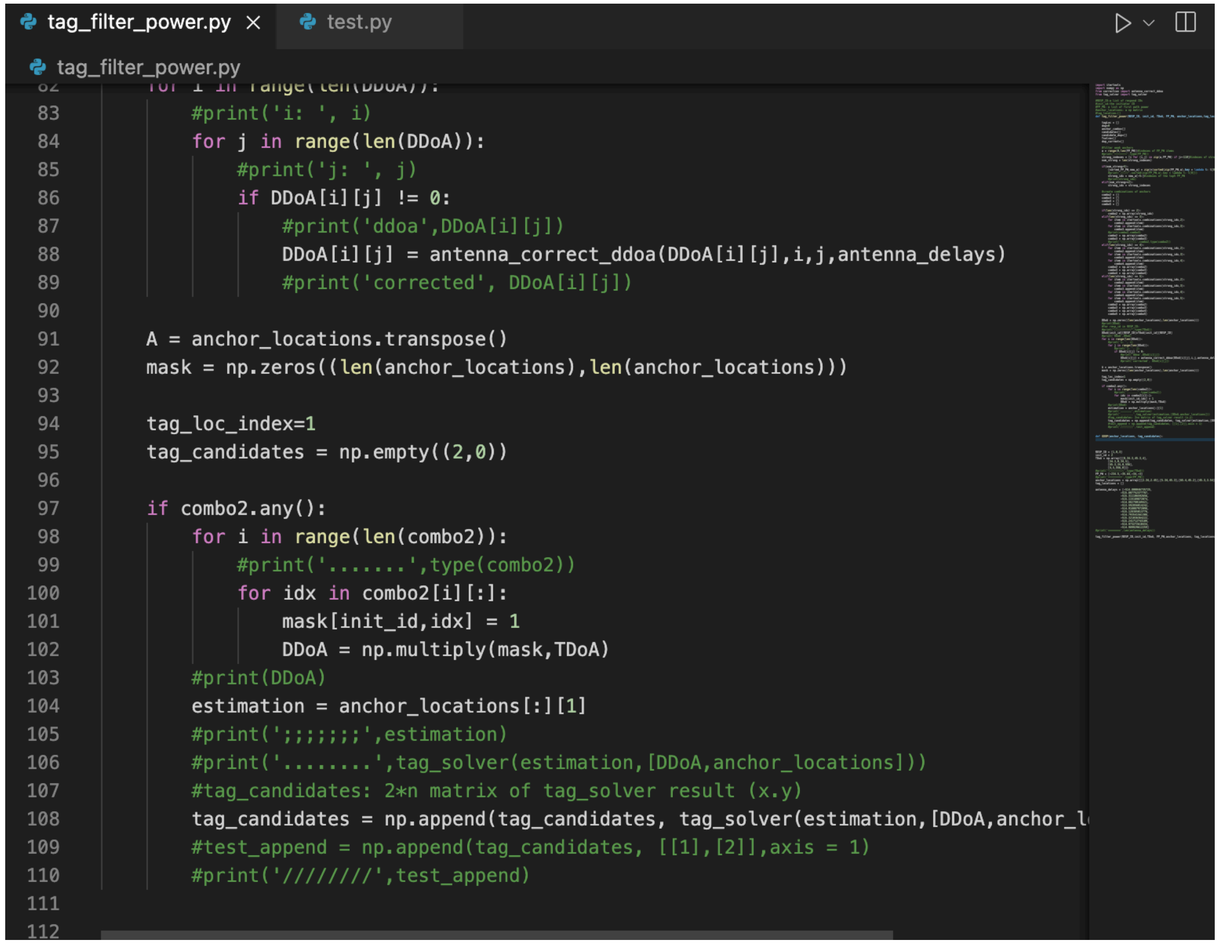
- Tag solver works properly and gives tag location in correct matrix format

```
28      print(np.transpose(np.vstack((result,temp_result.x))))
29
30  anchor_locations = [[2.34,2.45],[5.34,45.3],[65.4,45.2],[45.3,3.54]]
31  DDoA = [[0,34.5,34.3,55.34],
32          [34.5,0,4.56,67.543],
33          [34.3,4.56,0,4.67],
34          [55.34,67.543,4.67,0]]
35  estimation = [2.34,2.45]
36  tag_solver(estimation,[DDoA,anchor_locations])
37
```

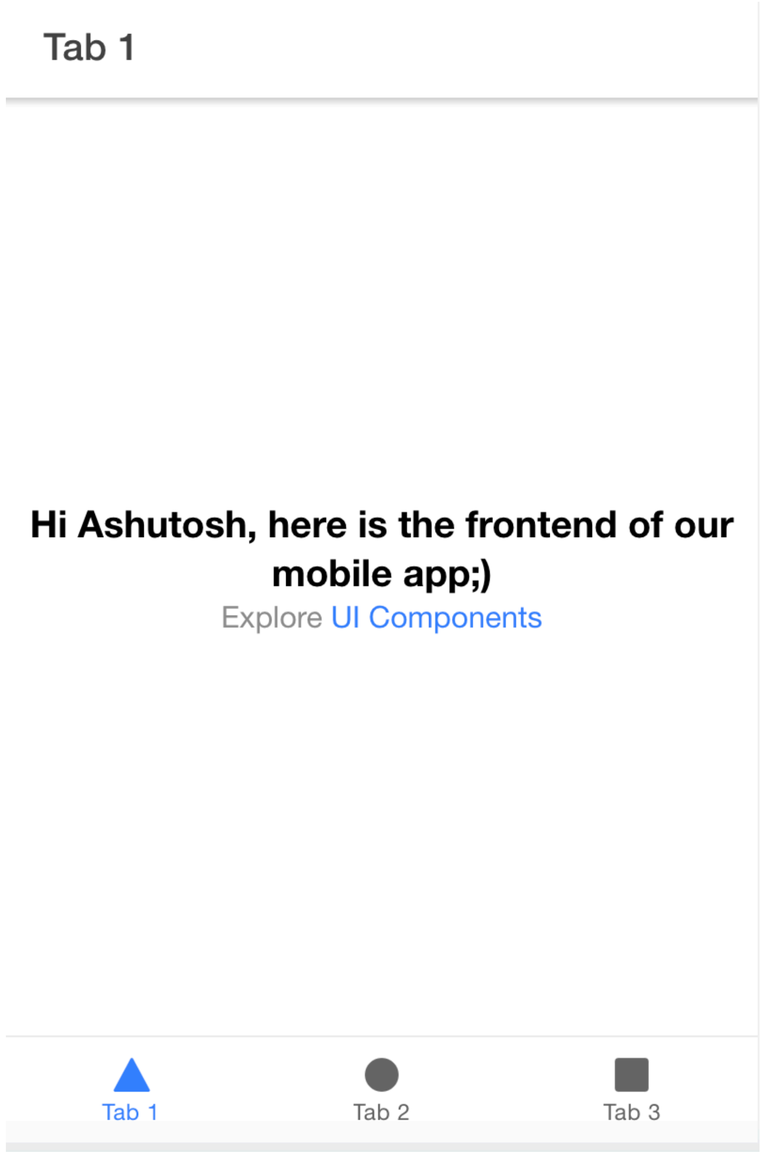
```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

yinzixin@lawn-143-215-90-164 indoorbackend % python tag_solver.py
[[ 31.338956  ]
 [ 29.43869181]]
yinzixin@lawn-143-215-90-164 indoorbackend %
```

Backend Code (tag filter, anchor selection)

- Anchor selection and tag filtering in progress;
- Been testing result for each block while coding



Frontend Code




```
<ion-header [translucent]="true">
  <ion-toolbar>
    <ion-title>
      Tab 1
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true">
  <ion-header collapse="condense">
    <ion-toolbar>
      <ion-title size="large">Tab 1</ion-title>
    </ion-toolbar>
  </ion-header>

  <app-explore-container name="Hi Ashutosh, here is the frontend of our mobile app;)"></app-exp
</ion-content>
```

```
//This is out send-receive data format.  We can just replicate this model for every type of c
export class RequestsService {
  //Specify a URL (or ip:port)
  base_path = 'http://localhost:8081';
  constructor(private http: HttpClient) { }

  //This is our very first method to perform a POST request
  sendPostRequest(coors) {
    //Angular expected headers
    const requestOptions = {
      headers: new HttpHeaders({
        'Content-Type':  'application/json'
        })
      };

    //Our payload data that will be sent to the server (gotten from the phone)
    let postData = {
      "pkt":"12345",
      "anchor_id":"123456",
      "fppw":999999999999,
      "TDOA":9999999999999999
    }
    //the Angular post method (URL , payload, http headers)
    this.http.post("http://localhost:8081/coors", postData, requestOptions)
      .subscribe(data => {
        //should receive data here
        console.log(data);

      }, error => {
        console.log(error);
      });
  }
```

Server Code

- Server.js calculate tag location and send back to frontend.

Server

```
var cors = require('cors')
var express = require('express');
var app = express();

app.use(cors())
app.use(express.json())


// This responds a POST request for tag solver
app.post('/tagsolver', function (req, res) {
   //receive payload from cellphone
   input = req.body
   //calculate tag locations (call from python code)
   tag_location = tagSolver(input);
   //send back calculated tag location
   res.send(tag_location);
})
```

Frontend

```
//This is out send-receive data format.  We can just replicate this model for every type of
export class RequestsService {
  //Specify a URL (or ip:port)
  base_path = 'http://localhost:8081';
  constructor(private http: HttpClient) { }

  //This is our very first method to perform a POST request
  sendPostRequest() {
    //Angular expected headers
    const requestOptions = {
      headers: new HttpHeaders({
        'Content-Type':  'application/json'
        })
      };

    //Our payload data that will be sent to the server (gotten from the phone)
    let postData = {
      "pkt":"12345",
      "anchor_id":"123456",
      "fppw":999999999999,
      "TDOA":9999999999999999
    }
    //the Angular post method (URL , payload, http headers)
    this.http.post("http://localhost:8081/solver", postData, requestOptions)
      .subscribe(data => {
        //should receive data here
        console.log(data);

      }, error => {
        console.log(error);
      });
  }
```