uOEC 2017 – Programming Challenge Problem Statement

THEME

The theme of the Programming Competition is "Accurate Positioning". Accurate Positioning incorporates the overall theme of the "Challenging the Greatest Engineering Innovation of 2016".

SCENARIO

The Greatest Engineering Innovation of 2016's main goal is to optimize human resources and time. This includes ThyssenKrupp's Multiway Elevator system, Alptransit Gotthard Base Tunnel, and TEB Technology Transit Elevated Bus [1]. It would be ideal to further develop these innovations, by allowing the observation of surroundings while working autonomously. In order for that to occur, the use of Simultaneous Localization and Mapping, SLAM is necessary [2].

The challenge is to utilize SLAM with Google's Cartographer or an equivalent detection software. More specifically, to design and construct a mobile application that uses an detection software to aid a lost uOttawa student to navigate around SITE. The hope is that this process will be used in a future mapping app for the inside of the school due to the accuracy the software provides.

OBJECTIVES, REQUIREMENTS & CONSTRAINTS

Objective

Develop an Application that provides real-time simultaneous localization and mapping (SLAM) in 2D or 3D. The application must display the area and detect the distance moved in a given area.

For example, say a user needs directions to the Tim Hortons in SITE. Cartographer is able to detect the distance and path accurately from any given position. Every position is different, so directions given from the Cartographer would be different from the front doors in SITE versus the stairs directly beside it.

The goal of the application is to navigate any point through SITE by building a map using Cartographer. The application should give live feedback when a point of interest such as a door or room number appears. When the user needs directions to a certain location, they should also be able to input a room number. Then the application will give them a notification if the room number is close to user or give them suggestions for where the user should move based on the mapping the application returns.

Requirements

1. Build ROS Cartographer on computer. The language used for Cartographer is a standalone C++ library, using ROS.
    - If your team decided to use alternate detection software for the challenge, a formal requested must be made to the uOEC committee and addressed to competitions@uottawaess.ca and to the category official with an alternate detection software.
2. Applications must be developed using Android Studio. You may use any industry standard language.
3. Application is allowed to include Google Cartographer open source libraries and resources.

4. Camera must be used as the only sensor used in this challenge to create the mapped area.
5. The Application doesn't need to be completely feasible to win this competition, implementation of the sensor using a detection software and Android Studio is the main aspect of the challenge.
6. The application must utilize basic Software Engineering practices focusing on the software's quality and user interaction.
7. Functional Requirements include:
    - The application must display the current location of the user.
    - The application must propose a route from the current location to the desired destination which can be entered by typing the building name or the building acronym.
    - You may add more functional requirements as you wish, the more requirements you put in, the more points you will receive.

After the development process, the application must be handed to the commissioner at the end of development phase. Presentations will follow the next day. Each team is required to present a 10 to 15 minutes presentation and demo the prototype, followed by a Q&A section. During the Q&A section, the judge will ask you questions. Your ability to answer them will contribute to your score as well. Your demo must include a runthrough of the application, and an example scenario of the tool locating the SANDBOX - STE 3041.

EXAMPLES, RESOURCES & REFERENCES

Examples of Applications using Cartographer

Galileo Offline Map
https://itunes.apple.com/app/galileo-offline-maps/id321745474?mt=8&ign-mpt=uo%3D4
Cartographer Map
http://cartographer-app.com/

Resources

[1] Cartographer ROS Integration
https://google-cartographer-ros.readthedocs.io/en/latest/index.html
[2] Android Studio and ROS
http://www.stratom.com/blog/2016/03/03/build-rosjava-library-sources-in-ros-w-android-studio
[3] Information on Google's Cartographer
https://opensource.googleblog.com/2016/10/introducing-cartographer.html
[4] Cartographer on GitHub
https://github.com/googlecartographer
[5] Getting Started with Cartographer
https://google-cartographer.readthedocs.io/en/latest/

References

[1] Greatest Engineering Innovation of 2016

http://www.popsci.com/11-greatest-engineering-innovations-year
[2] SLAM
http://www.computervisionblog.com/2016/01/why-slam-matters-future-of-real-time.html