Abstract

It is an iOS-ROS interface that connects the Corobots and iPhone device. This allows the ROS package to broadcast information such as Kinect video, network server information and Corobots status over the network.

The Google firebase provides authentication service for the iOS app. Once logged in, the user can localize his/her location using the QR codes found on the RIT CS floor. This location is then sent to web server.

Once the Corobots are live, the app is able to dispatch the Corobot and view its live Kinect stream. This app enables the user to view the status feed of the live Corobots around his/her location.

Architecture

The Corobot ROS system has several topics running and, for this project, we are interested in subscribing the Kinect “rgb/raw_image”. The ROS Package creates a web server to cast the Kinect live stream. This stream is directly streamed to the iOS device. The web server contains the overall status of all robots. This information is crawled by the iOS app which helps get the status of currently running Corobots and also the dispatch command.

The user of the app localizes himself/herself using the QR codes found on the CS floor. This would help the Corobot know the user location and find its way to the user. The Firebase provides no-sql realtime database and authentication for the users of the app.

Implementation

The iOS-ROS interface has a QR reader to localize the location of the user on the CS floor. The Firebase provides authentication services. The Corobot workspace contains a new package to transmit the Kinect data and other interesting statistics about the Corobots.

The app is able to deploy the Corobot by sending command to the Corobot web server at vhost1.cs.rit.edu

Features

- Localization
- Dispatch Call
- Status Feed
- Kinect Stream
- Authentication

References

http://wiki.ros.org/mjpeg_server
https://firebase.googleblog.com
https://vhost1.cs.rit.edu/status.php
https://vhost1.cs.rit.edu/location.php