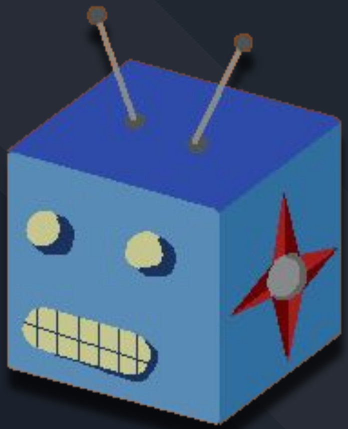


NaviBot Systems: Thirty-Gallon Robot Part Deux Design Review #2

February 28, 2020

Diva Ferrell, Logan Behnke, Peter Aaron Giroux,
Benjamin Peterson, George Cadel-Munoz
Sponsored By: Dr. Michael Leverington
Mentored By: Scooter Nowak

R.A.T. - An Introduction

Team Members:

- Diva Ferrell: Team Lead
- Logan Behnke: Release Manager
- Aaron Giroux: Recorder
- Benjamin Peterson: Customer Communicator
- George Cadel-Munos: Architect

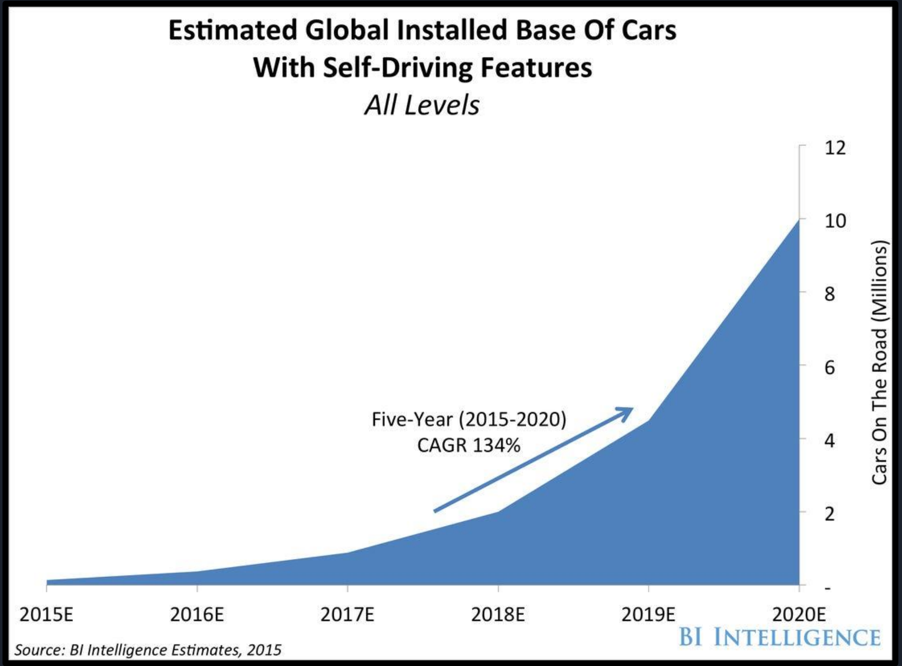
Sponsored by Dr. Michael Leverington

Mentored by Scooter Nowak

Problem Statement

Importance of Robotics and Navigation:

- Robotics present a potential to transform every aspect of our lives.
- Automation of historically human tasks is becoming more and more common.
- Navigation has become increasingly prominent in recent years.



Problem Statement

Dr. Leverington's idea for a Robot-Assisted Tour:

- Self-driving robot capable of guided tours throughout the Engineering Building.
- Relatively inexpensive components.
- Intended to serve as a recruitment tool for NAU.

NaviBot Systems' role:

- Implement a navigation module for R.A.T.
    - R.A.T. must be able to travel to a given location on its own.
    - R.A.T. must be able to localize himself within the engineering building.
    - GUI capable of tracking R.A.T.'s location and status.

Solution Overview

Four main modules are proposed to solve R.A.T.'s lack of functionalities

- GUI: An interactive interface allowing communication to and from R.A.T.
- Navigation: The ability for R.A.T. to move and operate with collision avoidance
- Mapping: The ability to read and develop maps
- WiFi Localization: The ability to triangulate R.A.T.'s location using WiFi signal strength

Requirements Review

Our requirements are obtained and refined through numerous processes:

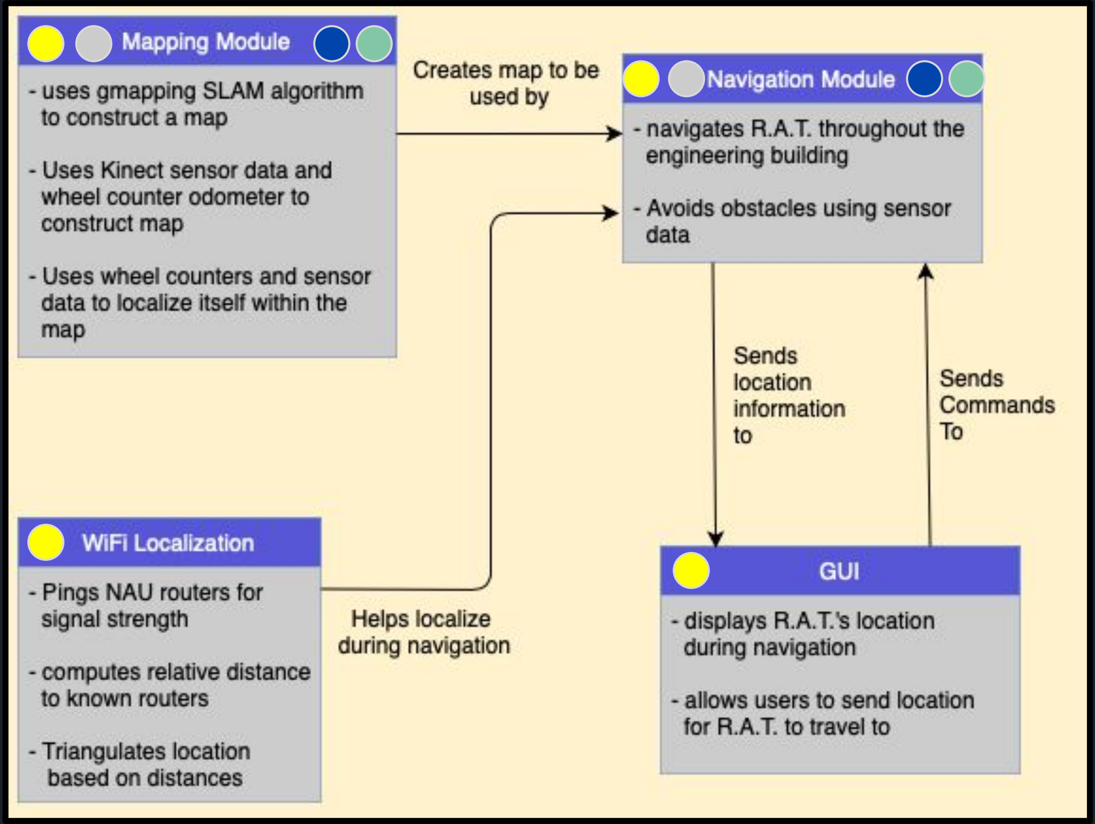
- Client Meetings
- Electrical Engineering Capstone Meetings
- Software and hardware requirements and capabilities

Finalization of requirements results in an implementation diagram with the following modules:

- GUI
- Mapping
- Navigation
- WiFi Localization

Implementation Overview

Language	Symbol
THE C PROGRAMMING LANGUAGE	
ROS	
C++	



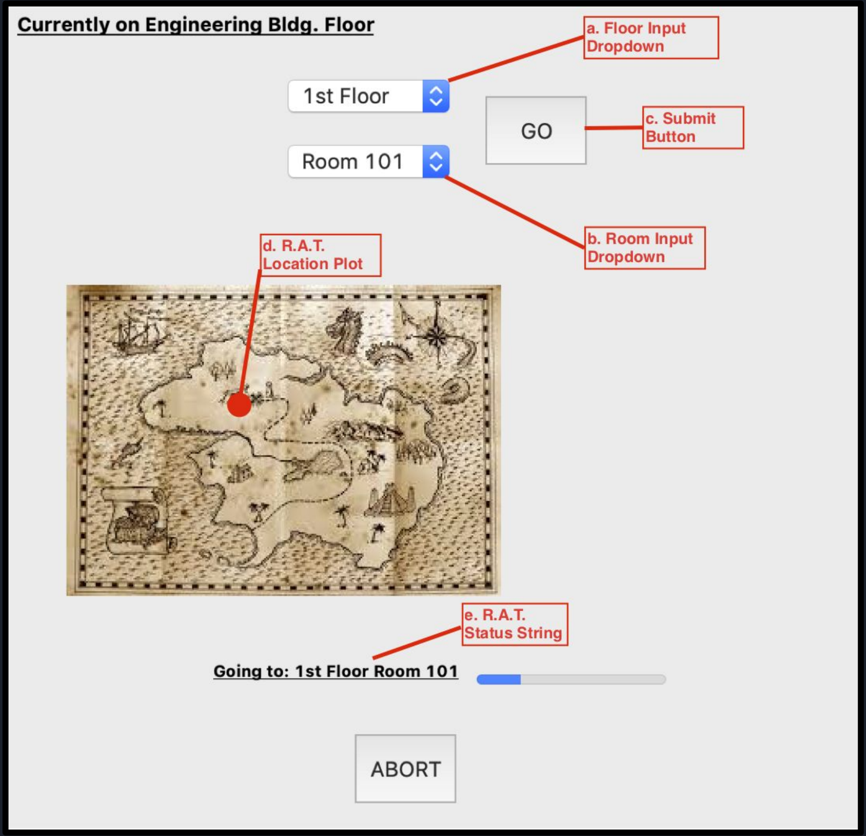
Architecture - GUI & Server

Displays R.A.T.'s location and status

- Updated every 15 seconds.
- Plots location on map of R.A.T.'s current floor.

Takes a floor and room number from a user

- Sends this location to R.A.T.
- R.A.T. begins navigation.



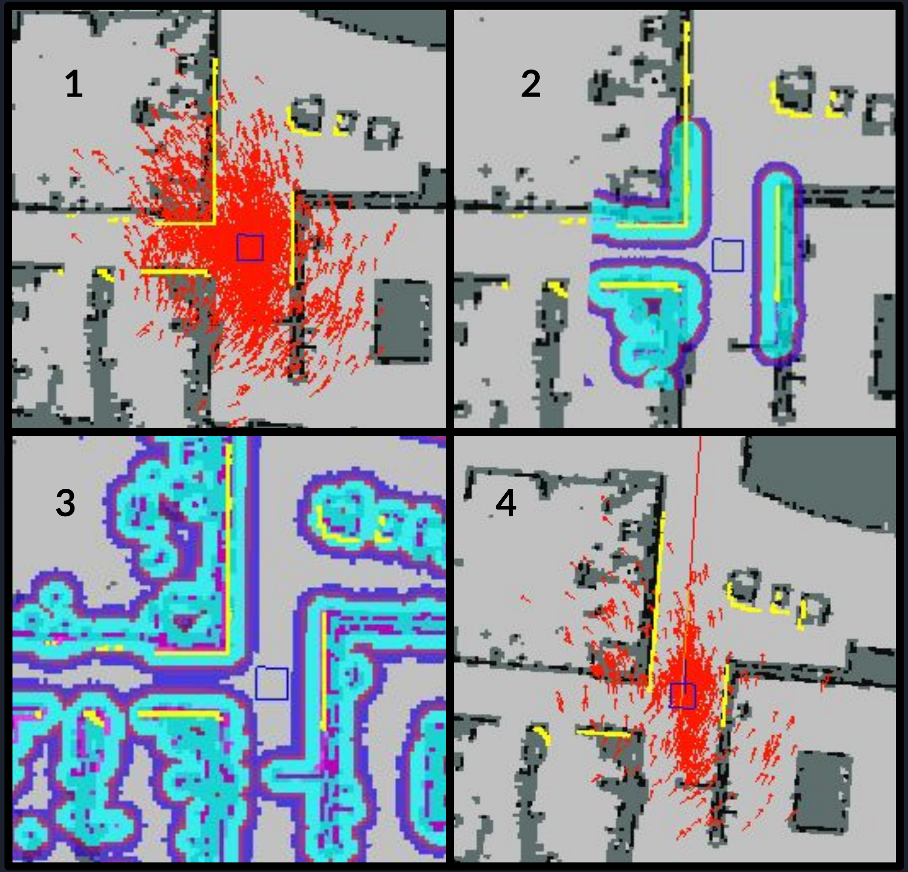
Architecture - Navigation

Localizes based on sensor data

Generate local costmap from the sensor data

Generate global costmap from the supplied map

Generate path based on costmaps



Architecture - Mapping

Generate a map based on sensor data and odometry data

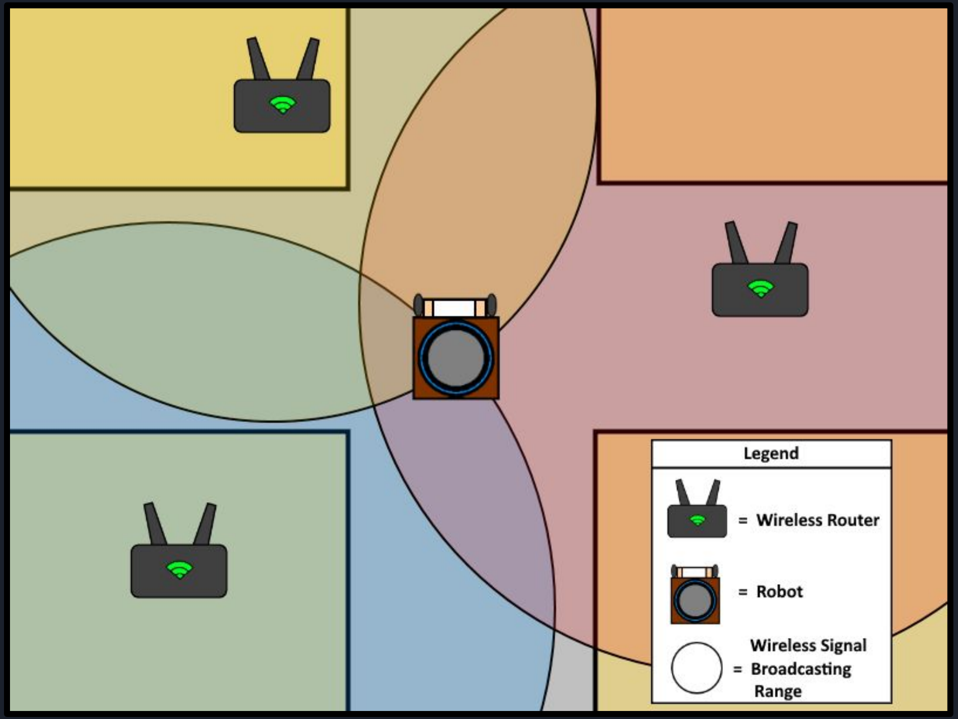
Architecture - WiFi Localization

Scans nearby broadcasting routers wirelessly

- Utilizes IEEE 802.11n protocol
- Collects signal strengths and MAC addresses

Computes distances from physical router locations

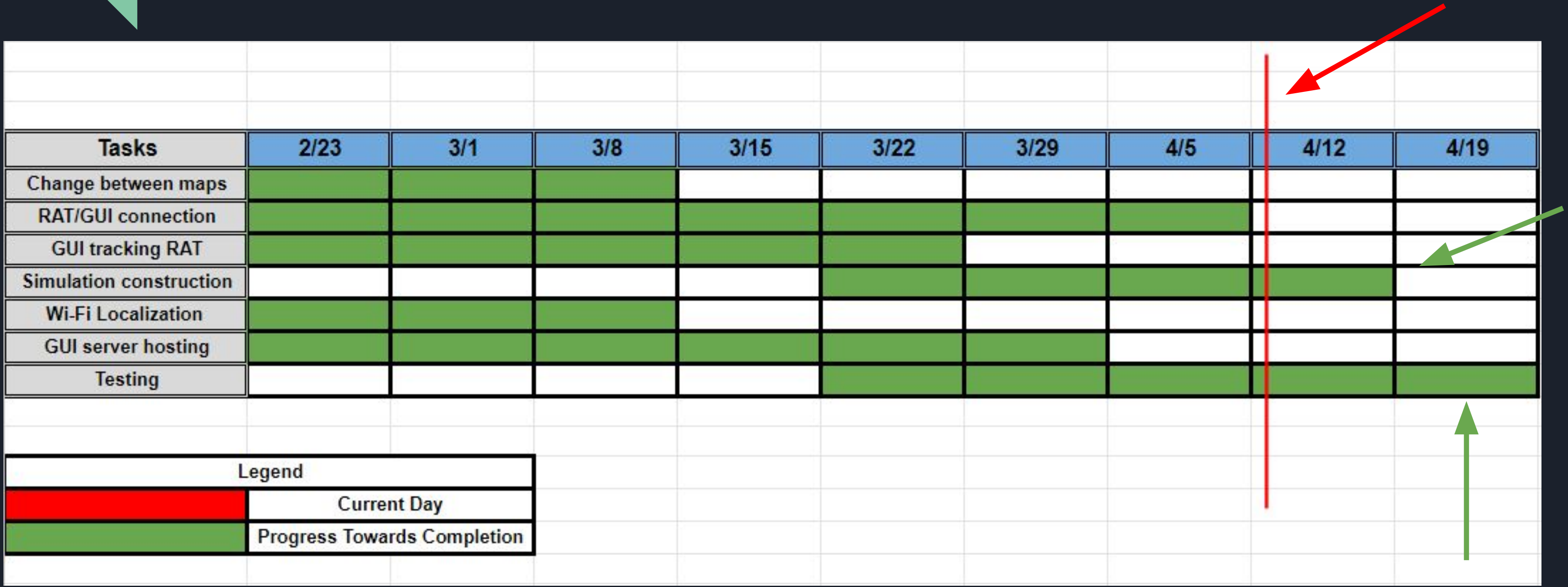
- Generates CSV file of stored information
- Calculates an approximate possible location of R.A.T.



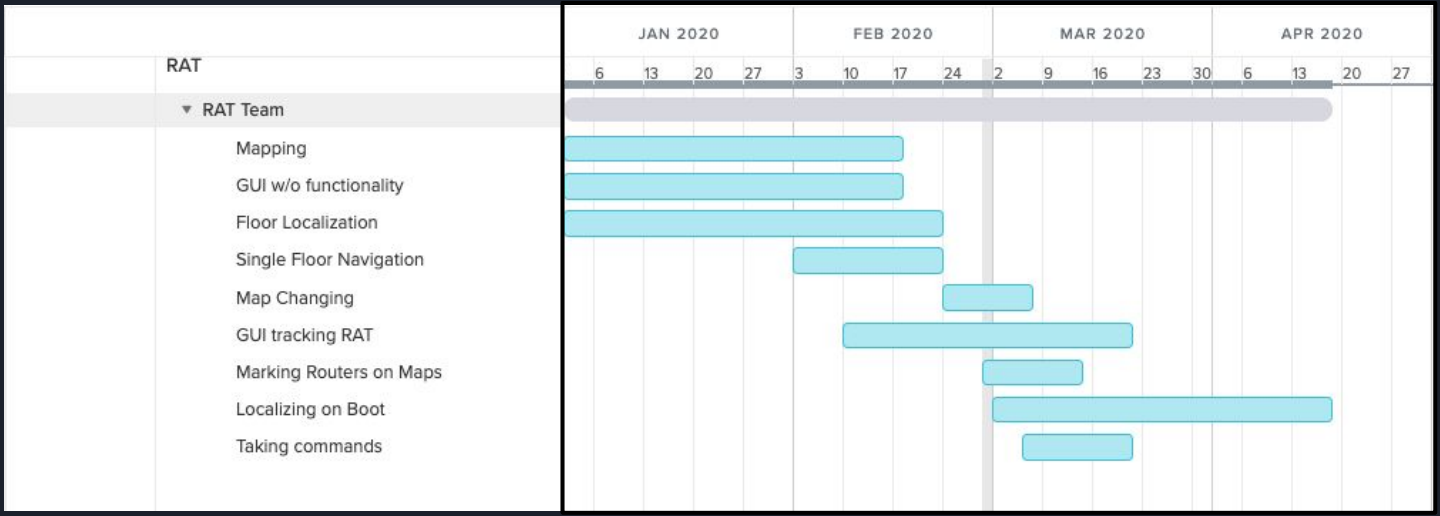
Challenges and Resolutions

Challenges	Solutions

Future Schedule



Future Schedule



Conclusion

- Robotics is an important and growing field
- Dr. Michael Leverington is looking for a cost effective robot to give tours of the engineering building
- Our main requirements include:
    - GUI
    - Mapping
    - Navigation
    - Localization
- GUI: An interactive interface allowing communication to and from R.A.T.
- Navigation: The ability for R.A.T. to move and operate with collision avoidance
- Mapping: The ability to read and develop maps
- WiFi Localization: The ability to triangulate R.A.T.'s location using WiFi signal strength

Thank you for Watching.

For more information please view our website:

https://tinyurl.com/NaviBotSystems

Or contact Diva Ferrell: djf222@nau.edu

Wi-Fi Loc: Signal Strength to Distance

Takes in a csv with Wi-Fi signal, Mac Address, frequency

Free Space Path Loss:

FSPL = Txp + Txa + Rxa - Rxs - FM

Distance Equation:

$$10^{\frac{(FSPL - K - (20 * log(f)))}{20}}$$

K = -27.55 for distance in meters and frequency in mHz

FSPL - Free Space Path Loss

Txp - Transmitter Power

Txa - Transmitter Antenna Gain

Rxa - Receiver Antenna Gain

Rxs - Signal Strength Received

FM - Fade Margin

K - constant based on units

f - frequency

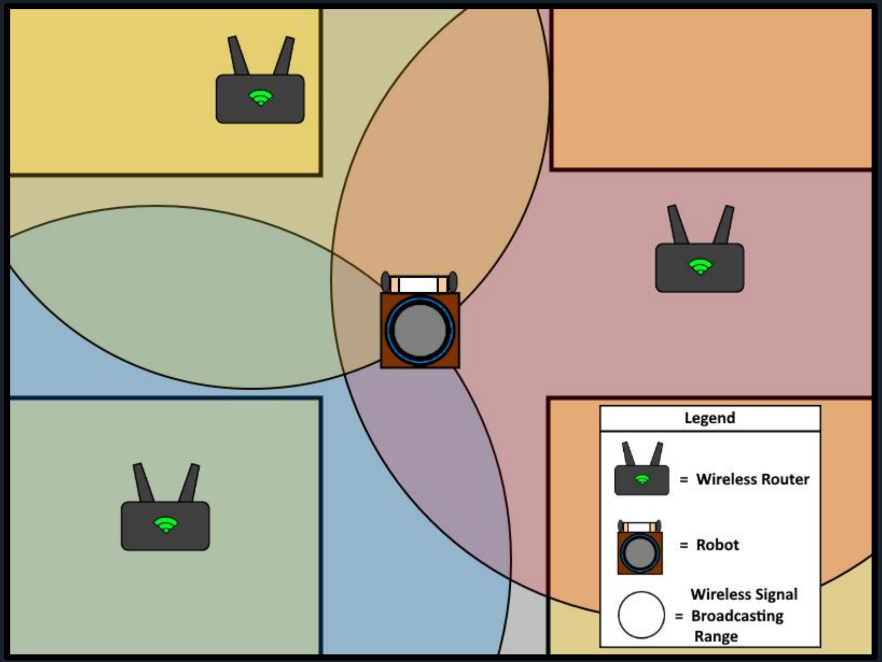
Wi-Fi Loc: Position Triangulation

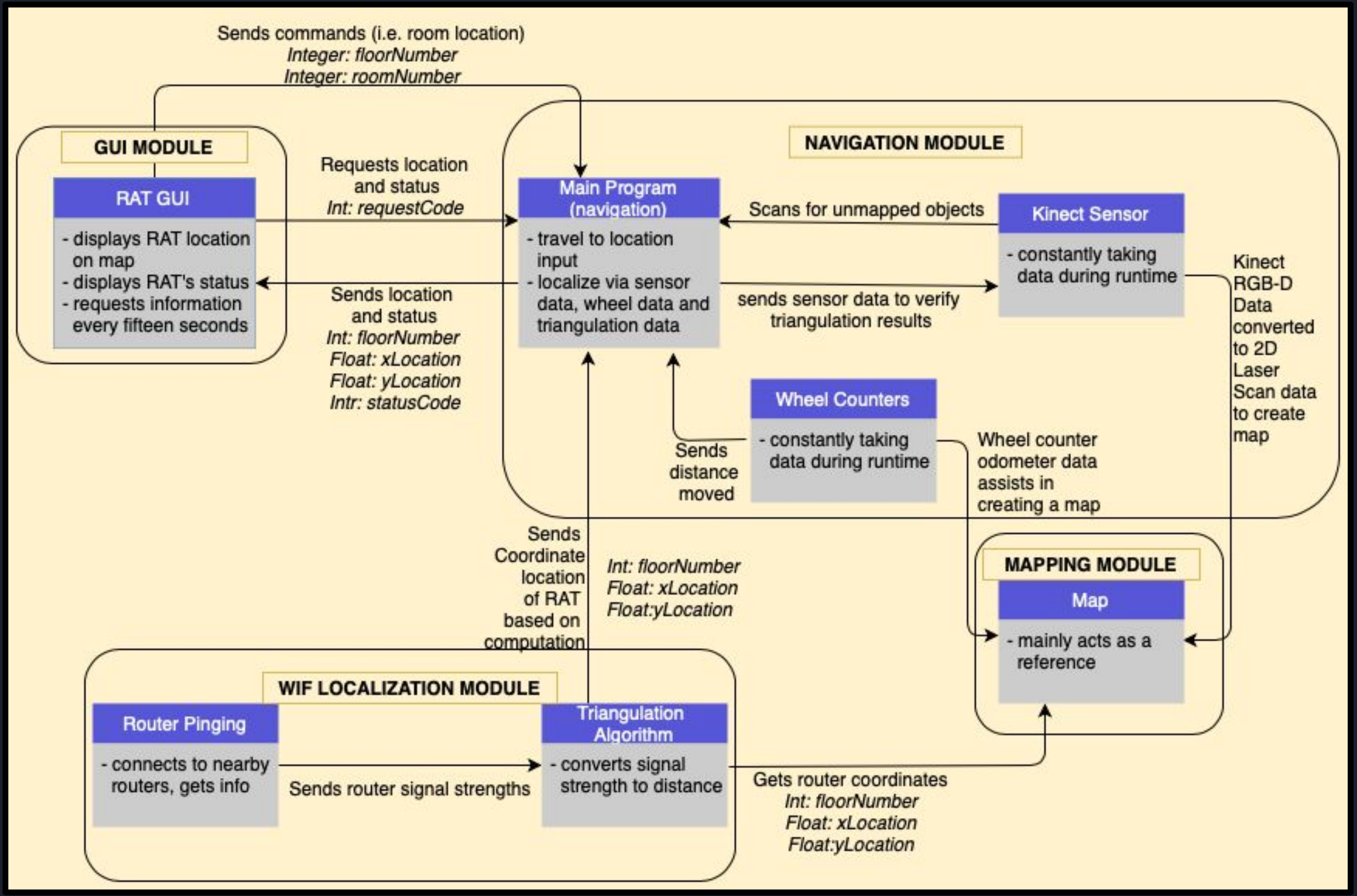
Use calculated distance as radius around each router

Use Pythagorean Theorem to calculate intersection points

Calculate midpoint of all intersections using weighted average

Send midpoint with margin of error to RAT navigation package




Sends commands (i.e. room location)
Integer: floorNumber
Integer: roomNumber
GUI MODULE
RAT GUI
- displays RAT location on map
- displays RAT's status
- requests information every fifteen seconds
Requests location and status
Int: requestCode
Sends location and status
Int: floorNumber
Float: xLocation
Float: yLocation
Intr: statusCode
NAVIGATION MODULE
Main Program (navigation)
- travel to location input
- localize via sensor data, wheel data and triangulation data
Scans for unmapped objects
Kinect Sensor
- constantly taking data during runtime
sends sensor data to verify triangulation results
Kinect RGB-D Data converted to 2D Laser Scan data to create map
Wheel Counters
- constantly taking data during runtime
Sends distance moved
Wheel counter odometer data assists in creating a map
Sends Coordinate location of RAT based on computation
Int: floorNumber
Float: xLocation
Float:yLocation
MAPPING MODULE
Map
- mainly acts as a reference
WIF LOCALIZATION MODULE
Router Pinging
- connects to nearby routers, gets info
Sends router signal strengths
Triangulation Algorithm
- converts signal strength to distance
Gets router coordinates
Int: floorNumber
Float: xLocation
Float:yLocation

GUI Architecture Diagram