Project Tips

Project Files

There are only two files you need to worry about for this project: `helpers.cpp` and `localizer.cpp`. Here's what each file included in the starter code is for:

- You can mostly ignore any `.h` (header) files, although if you add additional functions to `.cpp` files you should also define them within the related header file!
- The `maps` folder just has data of the map data used in the project
- `tests.cpp` is just for testing - no need to touch it
- `debugging_helpers.cpp` is to help you debug; don't need to implement anything
- `helpers.cpp` - Implement `normalize()` and `blur()`
- `localizer.cpp` - Implement `initialize_beliefs()`, `sense()` and `move()`
- `simulate.cpp` - You can uncomment portions of this to further help you develop the project, but this is not needed just to pass. See more tips below if you want to delve deeper here.

How to approach the TO-DO's

While there is no need to touch `tests.cpp`, it can help order your implementations, as you can unit test each function to see if it works before moving on further.

Therefore, the best method to approach the project is to write the code for one of these, compile your code (make sure to flag c++11!), then run `tests.cpp`. If that individual function passes the test without any errors, you can move onto the next one. If not, make sure to debug and fix it first!

Initializing vectors & matrices in C++11

In older versions of C++, you could initialize the size of the vector or matrix to start, but you had to either replace each index or use `.push_back()` to place values within the vector. In C++11, this is made easier as you can place the values to begin with, as shown below (assuming the various `some_vals` shown below are already initialized or are a `const`):