SELF-DRIVING OCEAN RESEARCH VESSELS TO MEASURE GLACIER RETREAT

Code Review Feedback and Responses

Team 46
Michael Gabriel
Donald ‘Max’ Harkins
Tobias Hodges
Chris Patenaude
Greg Sanchez

Reviewer: Dafei Du

Category	Reviewers Comment	Actions Take by Reviewed Group
Build	I was able to clone the repository and built everything using the README file.	No actions necessary
Legibility	The flow is very legible and clear. It is easy to read and understand the code because of useful comments and clear code style.	No actions necessary
Implementation	The implementation is perfect.	No actions necessary
Maintainability	There are several unit tests. Each test has comments to explain their purpose. It is very readable.	No actions necessary
Requirements	I think the code fulfills the requirements. They implement all features and have enough unit tests.	No actions necessary
Other	Group 46 did a great job.	No actions necessary

Reviewer: Kamron Ebrahimi

Category	Reviewers Comment	Actions Take by Reviewed Group
Build	Build guide contains all necessary information for configuring and setting up a local demonstration of the project. Unfortunately the dependencies are not lightweight as you need to install the entire Matlab runtime (2.3 GB). In addition a build guide is only provided for Linux machines and I was unable to build the project or run the tests on my Mac, even though it's Unix based. This does not seem like a serious issue because ultimately the project is intended to be run on a Linux machine with a very specific use case. I think spending the time to make the project build on each different operating system would be an enormous waste of time. Besides that I think it may be useful to include a brief section in the readme which explains what functionality is housed in which directories.	**A build guide is only provided for Linux machines and I was unable to build the project or run the tests on my Mac, even though it's Unix based** Our system is for an Intel NUC running Ubuntu 4.1.5. We currently do not have a use case for other operating systems, so we will not be adding further support for other platforms. **It may be useful to include a brief section in the readme which explains what functionality is housed in which directories.** We found this change unnecessary since each directory has a descriptive name which hints what functionality it contains. For example the "object_detection" dir contains all code related to object detection and classification.
Legibility	Code is legible and follows consistent structural formatting. In addition classes are properly scoped and intuitive to understand. My only critique on the legibility of the code would be perhaps the overuse of comments in some files. Generally comments should explain why some block of code exists and not what	**Overuse of comments in some files.** We intentionally used liberal comments knowing there would be another team next year who would pick up where we left off. We believe the descriptive comments we have in place will help them understand the code base and improve upon it in a timely manner.

	each line of code is doing. I understand your team is trying to make things easy to follow for reviewing group members but self documenting code (which I think your team produced in first place) is better than large comment blocks.	
Implementation	Implementation looks great. I will be taking inspiration from your unit testing method as my project also performs image processing. Stashing some test images in the Git repo is a great way to provide some test data for image processing algorithms.	No actions necessary
Maintainability	Code looks very easy to maintain, though as mentioned earlier I think placing greater emphasis on why functions and code blocks exist is more important than commenting every line of code with an explanation of what it does. Beside this the code is very elegant and clean. Testing infrastructure is also present with all the necessary mocks and automation scripts to validate that the system is working.	No actions necessary
Requirements	Obviously your project was heavily disrupted by Coronavirus as I recall your team mentioning that this term you would be using the software on the actual AVR. Hopefully something can still be Action Taken by Reviewed Groupworked out because my experience with working on systems is that things break during integration testing. Judging by the code and documentation it looks like your team gave your best effort to satisfy the requirements and build a robust solution. I will be very curious to hear what your team has to say during virtual expo on the	No actions necessary

	performance of the system if live testing on the AVR does occur this term.	
Other	No Comments	No Actions necessary

Reviewer: Nicholas Kiddle

Category	Reviewers Comment	Actions Take by Reviewed Group
Build	I was not able to build this project completely, however this was due to needing the necessary hardware. I followed the group's readme up to the final part where I was unable to specify the path to my IMU device since I did not have one. One problem I had that was not noted in the readme was a permission issue on /usr/local/MATLAB/R2020a/extern/engines/python/setup.py which prevented me from running the file. I think it would be worth adding a note to the readme about this. The last issue I noticed with the build was that the git repo has a lot of large files in it which makes it take a while to clone. In my experience it has been best to store large files in a shared drive and add an extra step to the readme. In this project this might include the example video and the weight matrices of the neural net being used. In the projects I have worked on it was standard practice to store replay video files and NN weights outside of the repo.	**I followed the group's readme up to the final part where I was unable to specify the path to my IMU device since I did not have one.** The project requires hardware to run as intended, and without an IMU, the project can only be run in test mode with predefined data. **One problem I had that was not noted in the readme was a permission issue on /usr/local/MATLAB/R2020a/extern/engines/python/setup.py which prevented me from running the file.** The permission issue with matlab permissions has been verified and handled resulting in the readme being updated accordingly. **The last issue I noticed with the build was that the git repo has a lot of large files in it which makes it take a while to clone. In my experience it has been best to store large files in a shared drive and add an extra step to the readme.** This is not something that we had a problem with but is something that we are considering moving forward. We recognize that including large files is not considered best practice for GitHub. However, the large file we have in our

	While this doesn't allow for the quickest build it makes pushing and pulling from the repo much easier and the downloads of large files are usually just a one time thing.	repository is a video needed to demonstrate system functionality for testing, and given the scope of the project we have decided to leave it in for the code freeze.
Legibility	The code is exceptionally legible. The code is very detailed in its comments. Honestly, the comments are almost too often that there is a gap between each line of code making it hard to just see the code. But I would not say that this is an issue since comments are always welcome. Each major component of the project has its own readme which is fantastic. I think that the documentation is very well broken up and will make it easy for the next group to pick this project back up. I would say that naming conventions would be useful. Some authors are using camel case and some are using underscores in their naming of variables and functions. All variable names seem to be descriptive enough and the amount of comments help. But using the same naming standards across the whole code base would help legibility and make it look cohesive.	**I would say that naming conventions would be useful. Some authors are using camel case and some are using underscores in their naming of variables and functions.** This issue has been addressed and the codebase now properly reflects PEP8 standards.
Implementation	The group is using primarily python which makes sense for the project. Python is very versatile. It is good for interfacing with hardware and data processing. Since the group needs to control the boat they will be able to make use of the extensive python libraries available to do so. Also the neural net they are using uses TensorFlow which is a well known machine learning package. I think that if they were to make improvements they could implement object oriented practices like classes. They could	**I think that if they were to make improvements they could implement object oriented practices like classes. They could have a boat class that has an IMU which has certain data and objects. The boat may also have motor controls and a camera class. I think this could help the code become more specific to each element and would help their workflow.** Our code base operates on a small set of data and transforms that data into a set of output vectors that is sent to the boat's OS and we believe the complexity of object oriented design does not suit the project scope.

	have a boat class that has an IMU which has certain data and objects. The boat may also have motor controls and a camera class. I think this could help the code become more specific to each element and would help their workflow.	
Maintainability	They have unit tests which test each major component of the project which will help the next group. There could always be more unit tests, but given the scale of this project I just don't think they have had the time to build a comprehensive testing suite. Like I mentioned earlier the amount of comments they have throughout their code will really help the next team to work on this project. It will be easy to pick up and get started. The functions they use are minimal and each have explicit purposes.	No actions necessary
Requirements	No. This project's requirements are a big stretch. The team was asked to build an autonomous boat from scratch. Getting a neural net working would be a couple of years of tuning alone since the conditions the boat would experience would always be fluctuating. Not to mention that due to COVID the group will not be able to work with the boat meaning not testing can happen. I do not see how they could have achieved the requirements given the situation. The group should push to perfect what they have now. Using the test data, they could perfect the tracking and image transformation and maybe optimize the code's computation efficiency. Optimization will be important during this project since the boat's autonomous controller will have to operate off a battery or generator so	This doesn't reflect our requirements document.

	lowering power usage will be necessary to run the machine for longer.	
Other	No comments	No Actions necessary

Reviewer: Unknown

Category	Reviewers Comment	Actions Take by Reviewed Group
Build	Repo can be cloned from GitHub. Projecet can be partially build just from ReadMe file.	No Actions necessary
Legibility	It is easy to follow. Code adhere to general guidelines.	No Actions necessary
Implementation	Functions look good. I don’t see any useful abstractions	No Actions necessary
Maintainability	Yes there is a unit test. It covers most of the cases.	No Actions necessary. We have since added more unit tests to cover a greater range of use cases.
Requirements	In my opinion code fulfill the requirements.	No Actions necessary
Other	Readme file can be improved by adding more detailed information	**Readme file can be improved by adding more detailed information** We had 3rd party reviewers go over our readme file following the instructions. We found there were areas in the text that could be clearer and rewrote those areas to be more explicit.