SCALE FOR PROJECT FT_KALMAN (HTTPS://PROJECTS.INTRA.42.FR/PROJECTS/FT_KALI

You should evaluate 1 student in this team

Git repository

```
git@vogsphere.42paris.fr:vogsphere/intra-uuid-374c405d-c999-49e1-a5da
```

Introduction

- Remain polite, courteous, respectful and constructive throughout the evaluation process. The well-being of the community depends on it.

- Identify with the person (or the group) evaluated the eventual dysfunctions of the work. Take the time to discuss and debate the problems you have identified.

- You must consider that there might be some difference in how your peers might have understood the project's instructions and the scope of its functionalities. Always keep an open mind and grade him/her as honestly as possible. The pedagogy is valid only and only if peer-evaluation is conducted seriously.

Guidelines

- Only grade the work that is in the student or group's GiT repository.

- Double-check that the GiT repository belongs to the student or the group. Ensure that the work is for the relevant project and also check that "git clone" is used in an empty folder.

- Check carefully that no malicious aliases was used to fool you and make you evaluate something other than the content of the official repository.

- To avoid any surprises, carefully check that both the evaluating and the evaluated students have reviewed the possible scripts used to facilitate the grading.

- If the evaluating student has not completed that particular project yet, it is mandatory for this student to read the entire subject prior to starting the defence.

- Use the flags available on this scale to signal an empty repository, non-functioning program, a norm error, cheating etc. In these cases, the grading is over and the final grade is 0 (or -42 in case of cheating). However, with the exception of cheating, you are encouraged to continue to discuss your work (even if you have not finished it) in order to identify any issues that may have caused this failure and avoid repeating the same mistake in the future.

- Remember that for the duration of the defence, no segfault, no other unexpected, premature, uncontrolled or unexpected termination of the program, else the final grade is 0. Use the appropriate flag.
You should never have to edit any file except the configuration file if it exists.
If you want to edit a file, take the time to explicit the reasons with the evaluated student and make sure both of you are okay with this.

- You must also verify the absence of memory leaks. Any memory allocated on the heap must be properly freed

before the end of execution.
You are allowed to use any of the different tools available on the computer, such as leaks, valgrind, or e_fence. In case of memory leaks, tick the appropriate flag.

Attachments

subject.pdf (https://cdn.intra.42.fr/pdf/pdf/129295/en.subject.pdf)

imu-sensor-stream-macos (https://cdn.intra.42.fr/document/document/25153/imu-sensor-stream-macos)

imu-sensor-stream-linux (https://cdn.intra.42.fr/document/document/25154/imu-sensor-stream-linux)

Mandatory Part

Prerequisite

- The code is written in C, C++ (standard C++17 or less) or Rust.
- The program doesn't use a filter library (only math libraries are allowed)
- The program compile statically (with a makefile if the language requires it)

Yes No

Transmissions

Launch the imu program with this command:"./imu-sensor-stream -d 90", and then launch the student filter. Repeat this at least 3 times.
Does it run until the end of the transmission? If not, try again multiple times. Only one fail out of 10 runs will be tolerated. Discuss the algorithm implementation during the transmissions.

Yes No

Bad signal

Try the filter with an impossibly noisier signal (i.e "./imu-sensor-stream -n 30.0") to see if the filter fails properly.

Yes No

Covariance matrix

Finally, ask the student to explain the purpose of his covariance matrix.

Yes No

Bonus Part.

Add bonuses ONLY if everything before is validated.

Does the program contain a trajectory visualization ?

1: Standard 2d plot
2: Standard 3d plot
3: 2d plot with variance display
4: 3d plot with variance display
5: 3d visualization with variance and nice aesthetics!

Rate it from 0 (failed) through 5 (excellent)

How fast is the filter ?

Run the filter on the imu launched with this command : "./imu-sensor-stream -d 10 --meanspeed".
1: Less than 0.005 second
2: Less than 0.001 second

3: Less than 0.0005 second
4: Less than 0.00015 second
5: Less than 0.00010 second

Rate it from 0 (failed) through 5 (excellent)

Does the filter work with a noisier signal ?

1: Goes through a 1 hour with 1.1 times more noise
2: Same with 1.2 times more noise
3: Same with 1.3 times more noise
4: Same with 1.4 times more noise
5: Same with 1.5 times more noise

Rate it from 0 (failed) through 5 (excellent)

Another bonus feature ?

Grading is at the discretion of the evaluating student.
By new feature we mean new functionality (adding colors or aesthetic-only features doesn't count as a bonus).

Rate it from 0 (failed) through 5 (excellent)

Ratings

Don't forget to check the flag corresponding to the defense

Ok Outstanding project

Empty work No author file W Invalid compilation Norme Cheat d Crash Leaks

l Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation