Introduction

Please comply with the following rules:

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

- Identify with the student or group whose work is evaluated the possible dysfunctions in their project. Take the time to discuss and debate the problems that may have been identified.

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

Guidelines

- Only grade the work that was turned in the Git repository of the evaluated student or group.

- Double-check that the Git repository belongs to the student(s). Ensure that the project is the one expected. 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 that is not the content of the official repository.

- To avoid any surprises and if applicable, review together any scripts used to facilitate the grading (scripts for testing or automation).

- If you have not completed the assignment you are going to evaluate, you have to read the entire subject prior to starting the evaluation process.

- Use the available flags to report an empty repository, a non-functioning program, a Norm error, cheating, and so forth.
In these cases, the evaluation process ends and the final grade is 0, or -42 in case of cheating. However, except for cheating, student are strongly encouraged to review together the work that was turned in, in order to identify any mistakes that shouldn't be repeated in the future.

- 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 maps.zip fdf_linux minilibx_mms_20191025_beta.tgz

minilibx-linux.tgz minilibx_macos_sierra_20161017.tgz sources.tgz fdf

Preliminary tests

Minimal requirements

Does the assignment meet the minimal requirements?

- The repository isn't empty.
- Norminette shows no errors.
- No cheating.
- No forbidden function/library.
- The code compiles with the required options.
- The executable is named as expected.
- During execution, there is no brutal or unmanaged crash (segfault, bus error, and so forth).
- No memory leaks.

Yes No

Mandatory part

Error management

Test fdf without parameters, with too many parameters, a non-existing file or on which you have no rights. If those tests are passed, then it's all good. This is the only error management that is required. For now on, the maps inside the input files have to be formatted properly.

Yes No

Graphic management

Run the program with the 42 map provided from the project page and verify that:

- A window opens
- Something is drawn in the window
- You can see a isometric projection of the 42 map
- Pressing 'ESC' closes the window and exits the program in a clean way (no leaks).
- Clicking on the cross on the window's frame closes the window and exits the program in a clean way (no leaks).

Yes No

Line tracing

- Use a flat map with nothing but 0, sized 4x4. We have a flat wireframe grid with a projection that is used to give a 3D concept.
- Same map, with 1 point at a different altitude. Check that the result corresponds and that the 3D effect is rendered.

Yes No

Heavy map

Check whether the program handles a bigger map and an aleatory 16x16 map.

Yes No

Heavier map

Test with bigger, heavier maps. Either those provided in the intranet, the evaluated student, or your owns. Be logical and keep in mind the requirements of the subject regarding what could contain your maps (colors for example).

Yes No

Graphic responsive

If the graphical representation stayed fluid and pleasant in the last test with heavy maps, then it's cool.

Yes No

MiniLibX images

Take a look at the code and check whether the student uses the images from the MLX to draw the image instead of putting pixels one by one. ;)

Yes No

Bonus part

A lot of nice extras.

Extra projection

Can the map be represented using another projection (such as parallel or conic)?

Yes No

Zooooooooom

Is there a way to zoom in and out using the keyboard or mouse?

Yes No

Translate

Is there a way to translate the projection using the keyboard or mouse?

Yes No

Rotation

Is there a way to rotate the projection using the keyboard or mouse?

Yes No

Be crazy

Give one more point if there is any additional bonus you consider is fine.
Creativity is an important point in your education and in the digital world.

Yes No

Ratings

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

Ok Outstanding project

Empty work Incomplete work Invalid compilation Norme Cheat

Crash Concerning situation Leaks Forbidden function