more risks of cheat, injustice, laziness, that will harm everyone's skills development. We do count on your maturity and wisdom during these remote defenses for the benefits of the entire community.

SCALE FOR PROJECT SO_LONG (/PROJECTS/SO_LONG)

You should evaluate 1 student in this team

Git repository

Introduction

Please respect the following rules:

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

-     Identify with the person (or the group) evaluated the eventualdysfunctions 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 yourpeers might have understood the project's instructions and the scope of its functionalities.

Guidelines

- Only grade the work that is in the student or group'sGiT repository master branch and last commit.

- 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.

youand make you evaluate something other than the content
of the official repository.

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

- If the evaluating student has not completed that
  particularproject yet, it is mandatory for this student to read
  the entire subject before starting the defense.

- Use the flags available on this scale to signal an empty
  repository,the non-functioning program, crash, norm error,
  invalid compilation, cheating, etc.

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

Attachments

☐ subject.pdf (https://cdn.intra.42.fr/pdf/pdf/24630/en.subject.pdf)

☐ minilibx_opengl.tgz (/uploads/document/document/3897/minilibx_opengl.tgz)

☐ minilibx_mms_20200219_beta.tgz
(/uploads/document/document/3898/minilibx_mms_20200219_beta.tgz)

Mandatory part

Executable name

Check that the project compiles well (without re-link) when you execute the `make`
command and that the executable name is `so_long`.
If not use the invalid compilation flag at the end of the scale.

Yes   No

Map reading

- use different maps.

- test with different line sizes.

Also, check that the program returns an error and exits properly when the configuration file is misconfigured (for example an unknown key, double keys, an invalid path..).

If not, the defense is over and use the appropriate flag incomplete work, crash...

Yes   No

Technical elements of the display

We're going to evaluate the technical elements of the display.
Check that the level is a valid representation of the map used as parameter.
-     A window must open at the launch of the
program.It must stay open during the whole execution.

-     Hide all or part of the window either by using
another windowor by using the screen's borders, then
minimize the windows and maximize it back. In all
cases, the content of the window must remain
consistent.

Yes   No

User basic events

In this section, we're going to evaluate the program's user
generated events. Execute the 3 following tests. If at least one
fails, this means that no points will be awarded for this
section. Move to the next one.

-     Click the red cross at the top left of the window.
Thewindow must close and the program must exit
cleanly.

-     Press the ESC key. The window must close and the
programmust exit cleanly. In the case of this test, we will
accept that another key exits the program, for example,
Q.

-     Press the four arrow keys (we'll accept WASD or
ZQSD keys) in the order of your liking. Each key press

movement).

Movements

In this section, we'll evaluate the implementation of the player's movement.
Execute the 5 following tests. If at least one fails, this means that no points will be awarded for this section. Move to the next one.

- The player's spawning position must be in accordancewith the map file.

- Press the arrows keys to move in every direction on the map.

- is the game "playable"?

Walls & Sprites

In this section, we'll evaluate the map representation.
Execute the following tests. If at least one fails, this means
that no points will be awarded for this section. Move to
the next one.

- The wall's texture is well placed and the player cannot go through it.

- The collectible's texture is well placed and the player can pick it by walking on it.

- The Exit texture is well placed and the player can finish the game by walking on it but only after picking
  everycollectible.

- The player texture is well placed and can move in every direction except into the walls.

Counter

In this section, we'll evaluate the gameplay elements.
Execute the following tests. If at least one fails, this means
that no points will be awarded for this section.

- There's a small counter displayed on the shell that counts how many movements the player did.

Yes   No

Image usage

The program use mlx_put_image_to_window and not mlx put pixel !

Yes   No

Bonus

We will look at your bonuses if and only if your mandatory part is excellent. This means that you must complete the mandatory part, beginning to end, and your error management must be flawless, even in cases of twisted or bad usage. So if the mandatory part didn't score all the points during this defense bonuses will be totally ignored.

Enemies enemy patrols that cause the player to lose in

case of contact.

Yes   No

Draw calls

There's some sprite animation.
You need to explain what it is and how you did it.

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

GUI

The counter of movement is rendered on the screen using texture or mlx_put_string.

Ratings

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

Ok   Outstanding project

Empty work   No author file   ᙎ Invalid compilation   Norme   Cheat   ᗡ Crash

Leaks   ⅃ Forbidden function

Conclusion

Leave a comment on this evaluation

⁄⁄

Finish evaluation