nalysann

(https://profile.intra.42.fr)

Remember that the quality of the defenses, hence the quality of the of the school on the labor market depends on you. The remote defences during the Covid crisis allows more flexibility so you can progress into your curriculum, but also brings 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 PISCINE OCAML (/PROJECTS /42CURSUS-PISCINE-OCAML) / RUSH 00 (/PROJECTS /42CURSUS-PISCINE-OCAML-RUSH-00)

You should evaluate 2 students in this team

★

Git repository

`git@vogsphere.msk.21-school.ru:vogsphere/intra-uuid-8b2774c1-92ad-44` 📋

Introduction

For the sake of this evaluation, we ask you to:

- Stay mannerly, polite, respectful and constructive during this evaluation. The trust between you and the 42 community depends on it.

- Bring out to the graded student (or team) any mistake he or she might did.

- Accept that there might be differences of interpretation of the subject or the rules between you and the graded student (or team). Stay open-minded and grade as honnestly as possible.

Guidelines

- You must grade only what is present in the graded student's (or team) repository.

Attachments

☐ subject.pdf (https://cdn.intra.42.fr/pdf/pdf/27345/en.subject.pdf)

Preliminaries

This section is dedicated to setup the evaluation and to test the prerequisits. It does not rewards points, but if something is wrong at this step or at any point of the evaluation, the grade is 0, and an approriate flag might be checked if needed.

Respect of the rules

- The graded student (or team) work is present in his or her repository.

- The graded student (or team) is able to explain his or her
work at any time of the evaluation.

- The general rules and the possible day-specific rules are
respected at any time of the evaluation.

- The Makefile does not relink.

     ⊘ Yes   ✕ No

Mandatory part

This part must be perfect to access to the bonus part.

Grid

Run the program. Can you see a 3x3x(3x3) grid?

     ⊘ Yes   ✕ No

Playing

Can you fill the grid with X's and O's using some kind of input?
Of course, error handling for faulty input must be perfect.
Remember that you must NOT test Ctrl-D.

     ⊘ Yes   ✕ No

1v1 game

Are there two players who write X and O's alternatively, with an
indication of whose turn it is?

     ⊘ Yes   ✕ No

Nested grids

Can one player win a grid? And in case of a tie, does the player
who put the ninth symbol in the grid win it?

     ⊘ Yes   ✕ No

Winning the game

Can the game be ended properly? (With either player winning or
the game ending in a draw)

     ⊘ Yes   ✕ No

Mandatory bonuses

This part must be entirely ignored if the mandatory points are not fully satisfied. This part consists in fundamental improvement features, which are also mandatory as a first step of bonuses but not essential to the project's validation.

IA

Is there a one-player mode versus an IA ? A stupid IA won't do;
it must try to win or to prevent you from winning.

       ⊘ Yes   ✕ No

Player names

Is there an option to let players choose a name? Of course,
error handling has to be perfect (empty names, both players
having the same name...)

       ⊘ Yes   ✕ No

Consecutive games

Is there an option at the end of the game to start a new one?

⊘ Yes   ✕ No

Bonus points

Bonus points

The bonuses are evaluated IF AND ONLY IF the mandatory points and mandatory bonuses are satisfied 100%. Each interesting and relevant feature earns one point. You can also include efficient code organization, optimized runtime performance, Xbox One compatibility...

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

○

Ratings

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

✔ Ok

■ Empty work   ● No author file   ☻ Invalid compilation   ♫ Norme   ▰ Cheat   �># Crash

⊘ Forbidden function

Conclusion

Leave a comment on this evaluation

[   ]

Finish evaluation