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

SCALE FOR PROJECT RUBIK (/PROJECTS/RUBIK)

Introduction

We ask you for the good progress of this evaluation to
respect the following rules:

- Be courteous, polite, respectful and constructive in all
situations during this exchange. The bond of trust between the
community 42 and you depends on it.

- Highlight to the person (or group) noted the
possible malfunctions of work done, and take the time
to discuss and discuss.

- Accept that there may sometimes be differences
of interpretation on the subject's requests or the scope of the
features. Stay open-minded about
the other's vision (is he or she right or wrong?), And write down the most
honestly possible. The pedagogy of 42 only makes sense if
the peer-evaluation is done seriously.

Guidelines
- You should only evaluate what is on the student / group's rendering GiT repository
.

- Make sure that the GiT repository is the one
corresponding to the student or group and the project.

- Meticulously verify that no malicious alias has been used
to mislead you and have you evaluate anything other than
the content of the official repository.

- Any meaningful script facilitating the evaluation provided by one of the two
parties must be rigorously checked by the other party to
avoid unpleasant surprises.

- If the correcting student has not yet done this project, it is mandatory for this student to read the subject in full before starting this defense.

- Use the flags available on this scale to report an empty rendering, non-functional, a cheat case, etc. In this case, the evaluation is completed and the final grade is 0 (or -42 in the special case of cheating). However, except cheating, you are encouraged to continue to exchange around the work done (or not done precisely) to identify the problems that led to this situation and avoid them for the next rendering.

Attachments

⬜ Subject (https://cdn.intra.42.fr/pdf/pdf/3538/rubik.en.pdf)

preliminaries

Preliminary instructions

First check the following:

- There is a rendering (in the deposit git)

- No cheating, the student must be able to explain his code.

- Author file, external libraries used to justify ...

If an element of this list is not respected, the notation stops there. Use the appropriate flag. You are encouraged to continue discussing the project, but the scale is not applied.

◌̌ Yes   ✕ No.

Mandatory part

Simple problems

Error management

Try the program without parameters, with too many parameters, with invalid parameters, with invalid sequences ... In short, try to put the program in default. Does the program have the expected behaviors (display of usage, leave with a message, etc.)?

⊘ Yes   ✕ No.

Simple position

Try the program with simple mixes, up to 4-5 movements only. Is the result correct? Is it displayed within a reasonable time (2-3 seconds)? If the answer to any of these questions is no, the defense stops there.

⊘ Yes   ✕ No.

Position a little more complex

Try the program with more complex mixes, between 5 and 20 movements. Is the result correct? Is it displayed within a reasonable time (2-3 seconds)? If the answer to any of these questions is no, the defense stops there.

⊘ Yes   ✕ No.

Random position

Try the program with random mixes, larger than 20 moves. Is the result correct? Is it displayed within a reasonable time (2-3 seconds)? If the answer to any of these questions is no, the defense stops there.

⊘ Yes   ✕ No.

Performance

For mixtures of sizes greater than thirty movements, the program is it able to solve on average in less than 150 movements and still within 2-3 seconds, including display?
If the answer is no, then put 0 to this question.
Add 1 point for each 20 gained moves.

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

◯

Distance-20

U R2 FBR B2 R B2 RU U2 L 'D' FR R2 'L B2 U2 F2
This position, called Superflip is part positions which always require at least 20 moves to be resolved ("Distance-20"). Does the program solve this Cube in a number of moves comparable to the results of the previous questions? Try other mixtures of this type (http://www.cube20.org/distance20s/).

☑ Yes ✕ No.

bonus
Bonus game

Program bonus

The bonus ladder is at the free judgment of the marker. A really hardcore bonus deserves more than two really easy bonuses. For example, among the super-class bonuses, which should be worth a lot of points if they are correctly made:
- Presence of a real 3D graphical interface that debunks (for example that shows the resolution in a piecemeal fashion)
- An algorithm really nag that goes down to the borders of what is possible (an average less than 30 movements, always in seconds, it's really very very very good and clearly deserves a big bonus)
- A choice between several resolution algorithms, even the automatic selection of the best of these algorithms as appropriate
- A program that accepts other n-size puzzles or odd shapes (2x2x2, 4x4x4, Megaminx, Square-1 ...)
- ...

In the bonuses really less hardcore but nevertheless useful, we can for example find:
- A generator of mixtures,
- A subdivision of the answer into stages "humanly understandable"
- ...

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

Bonus answer

Find a Rubik's Cube and demonstrate your agility. The more impressive it is, the better. Hint: Under 30-40 seconds, it's VERY impressive. But it's not just speed in life, be creative! And talk to each other ... :-)

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

ratings

Do not forget to check the flag

✔ Okay   ★ Outstanding project

📄 Empty work   📄 Incomplete work   💬 No author file   💀 Invalid compilation

📜 Standard   🗐 cheat   ☢ Crash   🚫 Forbidden function

Conclusion

Leave a comment on this evaluation

Preview !!!

General term of use of the website (https://signin.intra.42.fr/legal/terms/6)

Privacy policy (https://signin.intra.42.fr/legal/terms/5)

Legal notices (https://signin.intra.42.fr/legal/terms/3)

Declaration on the use of cookies (https://signin.intra.42.fr/legal/terms/2)

Terms of use for video monitoring (https://signin.intra.42.fr/legal/terms/1)

(https://sig