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

Menu

profile.intra.42.fr)

Menu

profile.intra.42.fr)

Holy Graph

rojects.intra.42.fr)

List projects (/projects/list)

earning.intra.42.fr)

Your projects

forum.intra.42.fr)

Company final evaluation

(/projects/first-internshipcompany-final-evaluation)

mpanies.intra.42.fr)

FdF (/projects/filler)

'meta.intra.42.fr)
First Internship

/shop.intra 22tfts (/projects/ft_ls)

(/projects/first-internship)

Matrice (/projects/matrice)





rnugroho

SCALE FOR PROJECT N-PUZZLE (/PROJECTS/N-PUZZLE)

You should correct 2 students in this tean



Git repository

vogsphere@vogsphere.42.fr:intra/2018/a



Introduction

In order to have a productive and tolerable grading session, we ask that you :

- Stay courteous, polite, respectful and constructive during this session.

The bond of trust between members of the 42 community depends on it;

- Take care to show the graded person(s) the problems you notice, and

explain them as best you can;

- Accept that there may be differences in interepretation on the featureset

and/or what the subject requires. Stay open-minded, try to honestly determine who is right and who is not, and grade accordingly.

Guidelines

Remember that you must ONLY grade what's on the turn-in repository!

You have to "git clone" the repository, and grade what's on it, AND ONLY WHAT IS ON IT.

Attachments

Subject (https://cdn.intra.42.fr/pdf/pdf/103/npuzzle.pdf)

Puzzle generator
(/uploads/document/document/99/npuzzle-
gen.py)

First and foremost

Preliminary checks

Check the following elements:

- There is something in the git repository
- The "auteur" file, if required by the subject, is present and valid
- The Makefile, if required, is present and has the required rules

If one of these elements is not in confirmity with what the subject requires, the session stops. You may still debate on the project, but you

are not to grade the student(s).

During the rest of this session, if the program has an inappropriate behaviour (Segfault, bus error, double-free, uncaught exception, etc ...),

the session stops.

✓ Yes

 \times No

Actually running the program

Output correctness

The students have the output required by the subject, which is:

- Complexity in time
- Complexity in size
- Number of moves from initial state to solution
- Ordered sequence of states that make up the solution

Or, if the puzzle is unsolvable, it has to be displayed, and the program has to exit cleanly.

✓ Yes

 \times No

Solution

Test the program on some puzzles, some from the students (the subject

requires that they bring some) and some using the generator

in the subject.

The program has to find a solution when it is possible, and if the puzzle is unsolvable, it has to tell the user and exit cleanly.

The solution has to be valid: Check that the program does not "cheat".

If the generator says the puzzle is solvable, then it IS solvable, so no excuses.

>>> IF YOU DO NOT ANSWER YES TO THIS QUESTION, THE SESSION ENDS, DO NOT GRADE THE REST. <<<

✓ Yes

 \times No

Bare-minimum time efficiency

The program is able to find a solution to a solvable 3-puzzle in under a few seconds. More than 10 seconds is unacceptable.

✓ Yes

 \times No

Search algorithm

A* algorithm

The students implemented the A^* algorithm or one of its variants (B^* , IDA*, etc...) AND are able to explain it comprehensibly.

✓ Yes

 \times_{No}

Heuristics

The students let the user choose between at least 3 (relevant) heuristic

functions AND are able to explain how their heuristics work AND are able

to explain why their heuristics are admissible.

✓ Yes

 \times No

Data structures 1

Open set

The students have implemented their "open" set with a relevant data structure.

- A priority queue (i.e a queue that gives you the high-priority elements first, here "high-priority" means "low-cost"), or another container that allows for immediate retrieval of the lowest-cost item, is worth 5
- A map/dict/hash of (cost) -> (node list) is worth 3
- Another container with a kind of mechanism to easily find the highest

priority element is worth 2

- Any other container, such as list/vector/etc ... is worth 0

Also, the students must be able to explain the relevance of their choice.

If they can not, this is worth 0



Closed set

The students have implemented their "closed" set with a relevant data structure.

- A container that easily allows to check whether a node currently is in the set or not is worth 5
- Any other container, such as list/vector/etc ... is worth 0

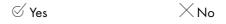
Also, the students must be able to explain the relevance of their choice. If they can not, this is worth 0



Data structures2

Greedy search 1

The students have implemented an option to do a greedy search and are able to explain it.



Greedy search2

	,	
	✓ Yes	×No
Other int	teresting bonu	ses
	•	ther interesting bonuses. 1 point pe u deem relevant and/or interesting
	Rate it from 0 (failed	d) through 5 (excellent)
_		
Rating		sponding to the defense
•		· ·
•	to check the flag corre	· ·
•	to check the flag corre	· ·
Don't forget	to check the flag corre	en function
Don't forget	to check the flag corre O Forbidd	en function