SCALE FOR PROJECT MATRIX (/PROJECTS/MATRIX)

You should evaluate 1 student in this team

★

Git repository

```
git@vogsphere-v2-bg.1337.ma:vogsphere/intra-uuid-66b12321-d
```

Introduction

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

- Identify with the person (or the group) evaluated the eventual dysfunctions 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 your peers might have understood the project's instructions and the scope of its functionalities. Always keep an open mind and grade him/her as honestly as possible. The pedagogy is valid only and only if peer-evaluation is conducted seriously.

Guidelines

- Only grade the work that is in the student or group's GiT repository.

- 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.

- Check carefully that no malicious aliases was used to fool you and make you evaluate something other than the content of the official repository.

- To avoid any surprises, carefully check that both the evaluating

and the evaluated students have reviewed the possible scripts used to facilitate the grading.

- If the evaluating student has not completed that particular project yet, it is mandatory for this student to read the entire subject prior to starting the defence.

- Use the flags available on this scale to signal an empty repository, non-functioning program, a norm error, cheating etc. In these cases, the grading is over and the final grade is 0 (or -42 in case of cheating). However, with the exception of cheating, you are encouraged to continue to discuss your work (even if you have not finished it) in order to identify any issues that may have caused this failure and avoid repeating the same mistake in the future.

- Remember that for the duration of the defence, no segfault, no other unexpected, premature, uncontrolled or unexpected termination of the program, else the final grade is 0. Use the appropriate flag.
You should never have to edit any file except the configuration file if it exists.
If you want to edit a file, take the time to explicit the reasons with the evaluated student and make sure both of you are okay with this.

- 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 (https://cdn.intra.42.fr/pdf/pdf/113267/en.subject.pdf)

display_macos.tar.gz (https://cdn.intra.42.fr/document/document/22657/display_macos.tar.gz)

display_linux.tar.gz (https://cdn.intra.42.fr/document/document/22658/display_linux.tar.gz)

Exercise 00 - Add, Subtract and Multiply

Complexity

Ask the student to justify the complexity of the functions. It must be at most O(n) in time and O(n) in space.

Check the use of forbidden mathematical functions (see the subject).

Yes No

Add

Check the behaviour of the vector addition with the following parameters:

- '[0, 0]' and '[0, 0]' give '[0, 0]'
- '[1, 0]' and '[0, 1]' give '[1, 1]'
- '[1, 1]' and '[1, 1]' give '[2, 2]'
- '[21, 21]' and '[21, 21]' give '[42, 42]'
- '[-21, 21]' and '[21, -21]' give '[0, 0]'
- '[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]' and '[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]' give '[9, 9, 9, 9, 9, 9, 9, 9, 9, 9]'

Check the behaviour of the matrix addition with the following parameters:

- '[[0, 0], [0, 0]]' and '[[0, 0], [0, 0]]' give '[[0, 0], [0, 0]]'
- '[[1, 0], [0, 1]]' and '[[0, 0], [0, 0]]' give '[[1, 0], [0, 1]]'
- '[[1, 1], [1, 1]]' and '[[1, 1], [1, 1]]' give '[[2, 2], [2, 2]]'
- '[[21, 21], [21, 21]]' and '[[21, 21], [21, 21]]' give '[[42, 42], [42, 42]]'

Feel free to perform more tests on your own

Yes No

Subtract

Check the behaviour of vector subtraction with the following parameters:

- '[0, 0]' and '[0, 0]' give '[0, 0]'
- '[1, 0]' and '[0, 1]' give '[1, -1]'
- '[1, 1]' and '[1, 1]' give '[0, 0]'
- '[21, 21]' and '[21, 21]' give '[0, 0]'
- '[-21, 21]' and '[21, -21]' give '[-42, 42]'
- '[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]' and '[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]' give '[-9, -7, -5, -3, -1, 1, 3, 5, 7, 9]'

Check the behaviour of matrix subtraction with the following parameters:

- '[[0, 0], [0, 0]]' and '[[0, 0], [0, 0]]' give '[[0, 0], [0, 0]]'
- '[[1, 0], [0, 1]]' and '[[0, 0], [0, 0]]' give '[[1, 0], [0, 1]]'
- '[[1, 1], [1, 1]]' and '[[1, 1], [1, 1]]' give '[[0, 0], [0, 0]]'
- '[[21, 21], [21, 21]]' and '[[21, 21], [21, 21]]' give '[[0, 0], [0, 0]]'

Feel free to perform more tests on your own

Yes No

Multiply

Check the behaviour of vector scaling with the following parameters:

- '[0, 0]' and '1' give '[0, 0]'

- '[1, 0]' and '1' give '[1, 0]'
- '[1, 1]' and '2' give '[2, 2]'
- '[21, 21]' and '2' give '[42, 42]'
- '[42, 42]' and '0.5' give '[21, 21]'

Check the behaviour of matrix scaling with the following parameters:

- '[[0, 0], [0, 0]]' and '0' give '[[0, 0], [0, 0]]'
- '[[1, 0], [0, 1]]' and '1' give '[[1, 0], [0, 1]]'
- '[[1, 2], [3, 4]]' and '2' give '[[2, 4], [6, 8]]'
- '[[21, 21], [21, 21]]' and '0.5' give '[[10.5, 10.5], [10.5, 10.5]]'

Feel free to perform more tests on your own

Yes No

Exercise 01 - Linear combination

Complexity

Ask the student to justify the complexity of the function. It must be at most O(n) in time and O(n) in space.

Check the use of forbidden mathematical functions (see the subject).

Yes No

Basic tests

Test the behaviour of linear combinations of vectors with the following parameters:

- 'linear_combination([Vector::from([-42., 42.])], [-1.])' gives '[42., -42.]'
- 'linear_combination([Vector::from([-42.]), Vector::from([-42.]), Vector::from([-42.])], [-1., 1., 0.])' gives '[0.]'
- 'linear_combination([Vector::from([-42., 42.]), Vector::from([1., 3.]), Vector::from([10., 20.])], [1., -10., -1.])' gives '[-62., -8.]'
- 'linear_combination([Vector::from([-42., 100., -69.5]), Vector::from([1., 3., 5.])], [1., -10.])' gives '[-52., 70., -119.5]'

Feel free to perform more tests on your own.

Yes No

Exercise 02 - Linear interpolation

Complexity

Ask the student to justify the complexity of the function. It must be at most O(n) in time and O(n) in space.

Check the use of forbidden mathematical functions (see the subject).

Yes No

Basic tests

Check the behaviour of the function with the following parameters:

- 'lerp(0., 1., 0.)' gives '0.'
- 'lerp(0., 1., 1.)' gives '1.'
- 'lerp(0., 42., 0.5)' gives '21.'
- 'lerp(-42., 42., 0.5)' gives '0.'
- 'lerp(Vector::from([-42., 42.]), Vector::from([42., -42.]), 0.5)' gives '[0.0] [0.0]'

Feel free to perform more tests on your own.

Yes No

Exercise 03 - Dot product

Complexity

Ask the student to justify the complexity of the function. It must be at most O(n) in time and O(n) in space.

Check the use of forbidden mathematical functions (see the subject).

Yes No

Basic tests

Check the behaviour of the function with the following parameters:

- '[0, 0]' and '[0, 0]' gives '0'
- '[1, 0]' and '[0, 0]' gives '0'
- '[1, 0]' and '[1, 0]' gives '1'
- '[1, 0]' and '[0, 1]' gives '0'
- '[1, 1]' and '[1, 1]' gives '2'
- '[4, 2]' and '[2, 1]' gives '10'

Feel free to perform more tests on your own.

Yes No

Exercise 04 - Norm

Complexity

Ask the student to justify the complexity of the functions. It must be at most O(n) in time and O(n) in space.

Check the use of forbidden mathematical functions (see the subject).

☑ Yes ☒ No

Euclidean norm

Check the behaviour of the function with the following parameter:

- '[0]' returns '0'.
- '[1]' returns '1'.
- '[0, 0]' returns '0'.
- '[1, 0]' returns '1'.
- '[2, 1]' returns '2.236067977'.
- '[4, 2]' returns '4.472135955'.
- '[-4, -2]' returns '4.472135955'.

Feel free to perform more tests on your own.

☑ Yes ☒ No

Manhattan norm

- '[0]' returns '0'.
- '[1]' returns '1'.
- '[0, 0]' returns '0'.
- '[1, 0]' returns '1'.
- '[2, 1]' returns '3'.
- '[4, 2]' returns '6'.
- '[-4, -2]' returns '6'.

Feel free to perform more tests on your own.

☑ Yes ☒ No

Supremum norm

Test the function with several different vectors. Each time, the function must return the component of the vector with the greatest value.

☑ Yes ☒ No

Exercise 05 - Cosine

Complexity

Ask the student to justify the complexity of the function. It must be at most O(n) in time and O(n) in space.

Check the use of forbidden mathematical functions (see the subject).

☑ Yes ☒ No

Basic tests

Check the behaviour of the function with the following parameters:

- '[1 0]' and '[0 1]' gives '0'
- '[8 7]' and '[3 2]' gives '0.9914542955425437'
- '[1 1]' and '[1 1]' gives '1'
- '[4 2]' and '[1 1]' gives '0.9486832980505138'
- '[-7 3]' and '[6 4]' gives '-0.5462677805469223'

Since the order of the parameters doesn't matter (the function is said to be commutative), the function must return the same result if you swap them.

Feel free to perform more tests on your own.

☑ Yes ☒ No

Exercise 06 - Cross product

Basic tests

Check the behaviour of the function with the following parameters:

- '[0 0 0]' and '[0 0 0]' gives '[0 0 0]'
- '[1 0 0]' and '[0 0 0]' gives '[0 0 0]'
- '[1 0 0]' and '[0 1 0]' gives '[0 0 1]'
- '[8 7 -4]' and '[3 2 1]' gives '[15 -20 -5]'
- '[1 1 1]' and '[0 0 0]' gives '[0 0 0]'
- '[1 1 1]' and '[1 1 1]' gives '[0 0 0]'

Feel free to perform more tests on your own. When giving two vectors to the function, imagine them creating a plane. Then, the function must return a vector that is orthogonal (perpendicular) to that plane.

Check the use of forbidden mathematical functions (see the subject).

☑ Yes ☒ No

Exercise 07 - Linear transform

Complexity

Ask the student to justify the complexity of the function. It must be at most O(n^3) in time and O(n^2) in space.

Check the use of forbidden mathematical functions (see the subject).

☑ Yes ☒ No

Basic tests

Check the behaviour of the function with the following parameter:

- '[[0, 0], [0, 0]]' and any vector of dimension two. The function must always return vectors with only zeros in it.
- '[[1, 0], [0, 1]]' and any vector of dimension two. The function must always return the same vector as given in parameter.
- '[[1, 1], [1, 1]]' and '[4, 2]'. The function must return '[6, 6]'.
- '[[2, 0], [0, 2]]' and '[2, 1]'. The function must return '[4, 2]'.
- '[[0.5, 0], [0, 0.5]]' and '[4, 2]'. The function must return '[2, 1]'.

Feel free to perform more tests on your own

☑ Yes ☒ No

Exercise 08 - Trace

Complexity

Ask the student to justify the complexity of the function. It must be at most O(n) in time.

Check the use of forbidden mathematical functions (see the subject).

☑ Yes ☒ No

Basic tests

Check the behaviour of the function with the following parameter:

- '[[0, 0], [0, 0]]' returns '0'
- '[[1, 0], [0, 1]]' returns '2'
- '[[1, 2], [3, 4]]' returns '5'
- '[[8, -7], [4, 2]]' returns '10'
- '[[1, 0, 0], [0, 1, 0], [0, 0, 1]]' returns '3'

Feel free to perform more tests on your own

Yes No

Exercise 09 - Transpose

Complexity

Ask the student to justify the complexity of the function. It must be at most O(n^2) (value assignments) in time and O(n^2) in space.

Check the use of forbidden mathematical functions (see the subject).

Yes No

Basic tests

Check the behaviour of the function with the following parameter:

- '[[0, 0], [0, 0]]' returns '[[0, 0], [0, 0]]'
- '[[1, 0], [0, 1]]' returns '[[1, 0], [0, 1]]'
- '[[1, 2], [3, 4]]' returns '[[1, 3], [2, 4]]'
- '[[1, 0, 0], [0, 1, 0], [0, 0, 1]]' returns '[[1, 0, 0], [0, 1, 0], [0, 0, 1]]'
- '[[1, 2], [3, 4], [5, 6]]' returns '[[1, 3, 5], [2, 4, 6]]'

Feel free to perform more tests on your own

Yes No

Exercise 10 - row-echelon form

Complexity

Ask the student to justify the complexity of the function. It must be at most O(n^3) in time and O(n^2) in space.

Check the use of forbidden mathematical functions (see the subject).

Yes No

Basic tests

Check the behaviour of the function with the following parameter:

- '[[0, 0], [0, 0]]' gives '[[0, 0], [0, 0]]'

- '[[1, 0], [0, 1]]' gives '[[1, 0], [0, 1]]'
- '[[4, 2], [2, 1]]' gives '[[1, 0.5], [0, 0]]'
- '[[-7, 2], [4, 8]]' gives '[[1, 0], [0, 1]]'
- '[[1, 2], [4, 8]]' gives '[[1, 2], [0, 0]]'

Feel free to perform more tests on your own

☑ Yes ☒ No

Exercise 11 - Determinant

Complexity

Ask the student to justify the complexity of the function. It must be at most O(n^3) in time.

Check the use of forbidden mathematical functions (see the subject).

☑ Yes ☒ No

Basic tests

Check the behaviour of the function with the following parameter:

- '[[0, 0], [0, 0]]' returns '0'
- '[[1, 0], [0, 1]]' returns '1'
- '[[2, 0], [0, 2]]' returns '4'
- '[[1, 1], [1, 1]]' returns '0'
- '[[0, 1], [1, 0]]' returns '-1'
- '[[1, 2], [3, 4]]' returns '-2'
- '[[-7, 5], [4, 6]]' returns '-62'
- '[[1, 0, 0], [0, 1, 0], [0, 0, 1]]' returns '1'

Feel free to perform more tests on your own

☑ Yes ☒ No

Explanations

Ask the student to explain:

- What happens when the determinant of a matrix is '0'.
- What the determinant represents geometrically in the vector space (ie, what happens after using the matrix for a linear transformation, and what does the determinant describe)

If they cannot explain it, the evaluation ends here.

Yes No

Exercise 12 - Inverse

Complexity

Ask the student to justify the complexity of the function. It must be at most O(n^3) in time and O(n^2) in space.

Check the use of forbidden mathematical functions (see the subject).

Yes No

Basic tests

Check the behaviour of the function with the following parameter:

- '[[1, 0], [0, 1]]' returns '[[1, 0], [0, 1]]'
- '[[2, 0], [0, 2]]' returns '[[0.5, 0], [0, 0.5]]'
- '[[0.5, 0], [0, 0.5]]' returns '[[2, 0], [0, 2]]'
- '[[0, 1], [1, 0]]' returns '[[0, 1], [1, 0]]'
- '[[1, 2], [3, 4]]' returns '[[-2, 1], [1.5, -0.5]]'
- '[[1, 0, 0], [0, 1, 0], [0, 0, 1]]' returns '[[1, 0, 0], [0, 1, 0], [0, 0, 1]]'

Feel free to perform more tests on your own. To check the result, you can multiply it by the matrix you gave as parameter and it must give (approximately) the identity matrix (However, avoid testing matrices that are not invertible).

Yes No

Exercise 13 - Rank

Basic tests

Check the behaviour of the function with the following parameter:

- '[[0, 0], [0, 0]]' returns '0'
- '[[1, 0], [0, 1]]' returns '2'
- '[[2, 0], [0, 2]]' returns '2'
- '[[1, 1], [1, 1]]' returns '1'
- '[[0, 1], [1, 0]]' returns '2'
- '[[1, 2], [3, 4]]' returns '2'
- '[[-7, 5], [4, 6]]' returns '2'
- '[[1, 0, 0], [0, 1, 0], [0, 0, 1]]' returns '3'

Feel free to perform more tests on your own

Check the use of forbidden mathematical functions (see the subject).

☑ Yes ☒ No

Explanations

Ask the student to explain what the rank of a matrix represents.

If they cannot explain it, the evaluation ends here. You can use the internet to check the answers.

☑ Yes ☒ No

Exercise 14 - Bonus: Projection matrix

Projection

Build several matrices with several FoVs (convert the value in radians before passing it to the function):

- 100 degrees
- 70 degrees
- 40 degrees

Then, test the matrices in the projection utility given in the attachements.

Also, try testing with several different combinations of near/far values (near must stay smaller than far) and different ratios (the default is 1).

A lower FoV must reduce the angle of view.

Changing the ratio must distort the image.

Different values of near and far must change the distance from the camera at which objects disappear from the screen.

Ask the student to explain what each component of the matrix represents.

☑ Yes ☒ No

Exercise 15 - Bonus: Complex vector spaces

Lots of tests

For this exercise, the student must have recoded all the previous functions (except for ex14), or used the generic structure of the code, to provide the use of complex numbers as scalars. The student should be able to explain how the operations of complex numbers work (geometrically).

Reminder of the rules for complex numbers:

- 'i^2 = -1'
- '(a + bi) + (c + di) = (a + c) + (b + d)i'
- '(a + bi) - (c + di) = (a - c) + (b - d)i'
- '(a + bi) * (c + di) = (ac - bd) + (bc + ad)i'
- '(a + bi) / (c + di) = ((ac + bd) + (bc - ad)i) / (c^2 + d^2)'

Test every function, but with complex numbers, and check that they behave correctly. The student that has done this bonus should probably provide tests for complex numbers in his executables, and show them along with the correction for the regular exercises, if they wish to gain time.

☑ Yes ☒ No

Ratings

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

✔ Ok

Empty work | Incomplete work | Invalid compilation | Cheat | Crash

Incomplete group | Leaks | Forbidden function | Can't support / explain code

Conclusion

Leave a comment on this evaluation

Finish evaluation