fl 0 **ncsomori** (https://pr

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

(https://projects.intra.42.fr/scale_teams/4227061/edit#)

ofile.intra.42.fr/)

ojects.intra.42.fr/)

arning.intra.42.fr/)

rflow.com/c/42network)

panies.intra.42.fr/)

neta.intra.42.fr/)

hop.intra.42.fr/)

Menu

- My projects (https://projects.intra.42.fr/)
- Holy Graph (https://projects.intra.42.fr/projects/g
- List projects (https://projects.intra.42.fr/projects/lis
- Available Cursus (https://projects.intra.42.fr/cursus_su

Your projects

- Day 02 (https://projects.intra.42.fr/projects/4 piscine-c-formation-piscine-php-day-02)
- Day 07 (https://projects.intra.42.fr/projects/4 piscine-c-formation-piscine-php-day-07)
- Day 08 (https://projects.intra.42.fr/projects/4 piscine-c-formation-piscine-php-day-08)
- Day 09 (https://projects.intra.42.fr/projects/4 piscine-c-formation-piscine-php-day-09)
- Rush00 (https://projects.intra.42.fr/projects/p php-rush00)
- ft_printf (https://projects.intra.42.fr/projects/ft
- init (https://projects.intra.42.fr/projects/in
- Piscine PHP (https://projects.intra.42.fr/projects/p php)

SCALE FOR PROJECT PISCINE (HTTPS://PROJECTS.INTRA.42.FR/PI PHP) / DAY 06 (HTTPS://PROJECTS.INTRA.42.FR/P PISCINE-C-FORMATION-PISCINE-PI

You should evaluate 1 student in this team

Git repository

git@vogsphere-v2.hive.fi:vogsp

Guidelines

You already know how peer2peer evaluation work by now and during this PHP Piscine, it works as any other Piscine projects you have evaluate so far. Take the time to discuss the assignments and to make sure that the evaluated has understood the notions of the day.

Attachments

- Subject (https://cdn.intra.42.fr/pdf/pdf/6644/d06.en.
- Transcript of the videos missing subtitles (https://cdn.intra.42.fr/pdf/pdf/10970/day06_transcrip
- Videos (https://elearning.intra.42.fr/notions/piscine-ph to-object-oriented-programming-with-php/subnotions)
- Resources (https://projects.intra.42.fr/uploads/document/documen

Preliminaries

There are a lot of exercises, you will have to manage your time accordin fastest and most efficient tests. Be careful, this evaluation form won't nec test, you have to come up with your own! As you most likely have create your own assignments, feel free to re-use them. Once an exercise is wror stop but you can still discuss the rest of the assignments.

The basics

- The whole team must be present.
- Only grade the work that is in the student or group's GiT repository.
- You must use Chrome for this defense.
- If there is nothing in the repository, the evaluation stops here. Discuss what went wrong and how to avoid facing the same situation tomorrow.

If one of the following points is not respected, the evaluation stops and the grade is 0.
- To use forbidden PHP parts, especially notions that are taught in d07 ar
- An output that differs consequently from the required output (incomplet computation, missing feature, etc).
- A public method or attribute without explanation. If there is an explana convinced then it's up to you to decide.
- Failing one of the following instructions:
• Only one unique Class per file.
• One file that includes the definition of a class cannot include any other `require_once` if necessary.
• A file containing a class must ALWAYS be named ClassName.class.php
• A class must ALWAYS be accompanied by a documentation file that M ClassName.doc.txt.
• The documentation of a class must ALWAYS be useful and correspond
• A class must ALWAYS have a static Boolean attribute called verbose.
• A class must ALWAYS have a static method called doc that returns the
• An exercise folder should contain the files from previous exercises, wh altered or new.

The following are **not** eliminatory:
• To have an output that differs in its formatting (one more or less space
• To have a method or an attribute that doesn't have exactly the same n semantic though.
• To solve the problem with a different algorithm than the one explainec the result is identical.

☐ Yes ☐

Ex00

The Color Class

• The Color Class must have 3 public integer attributes: `red`, `green` an to represent to components of a color.

• The Class constructor requires an array. An instance must be able to b value for the `rgb` key which will be split into three red, green and blue c passing a value for the red, green and blue keys which will directly repre components.

• Each of the values for the four possible keys will be converted into an

• The Color Class must have a `__toString` method which output match

• The Class must include a Boolean static attribute called `verbose` to co to the use of the Class. This attribute is initially False.

• If and o*nly* if the static attribute verbose is true, then the Class construc produce an output.

• The Class must have a static method called `doc` that returns the docun string. The content of the documentation must be read from a `Color.doc`

• The Class must have a method called `add` that allows you to add the c instance to the components of another instance argument. The resulting c

• The Class must have a method called `sub` that allows you to subtract t current instance to the components of another instance argument. The re

instance.

• The Class must have a method called `mult` that allows you to multiply current instance to an argument value. The resulting color is a new instan

• Running the `main_00.php` script generates an output like the one in th

☐ Yes ☐

Ex01

The Vertex Class

• The Vertex class must have 5 private attributes to represent `x, y, z,`

• The Vertex's color is always an instance of the Color Class from the pr

• The Vertex Class must have reading and writing accessors for its five a

• The Class' constructor is expecting an array. The following keys are re
- `x` : x axis coordinate, mandatory.
- `y` : y axis coordinate, mandatory.
- `z` : axis coordinate, mandatory.
- `w` : optional, by default it's 1.0.
- `color` : optional, by default it's a new instance of the color white.

• The Vertex Class must have a `__toString` method which output matcl

• The Class must include a Boolean static attribute called `verbose` to co to the use of the Class. This attribute is initially False.

• If and *only* if the static attribute verbose is true, then the Class construc produce an output.

• The Class must have a static method called `doc` that returns the docun string. The content of the documentation must be read from a `Vertex.do`

• Running the `main_01.php` script generates an output like the one in th

☐ Yes ☐

Ex02

The Vector Class

• The Vertor class must have 4 private attributes to represent `x, y, z` e

• The Vector Class must have read-only assessors for its four attributes.

• The Class' constructor is expecting an array. The following keys are re
- `dest` : the vector's destination vertex, mandatory.
- `orig` : the vector's origin vertex, optional, and which by default is a ne `y=0, z=0, w=1` vertex.

• The Vertor Class must have a `__toString` method which output matcl

• The Class must include a Boolean static attribute called `verbose` to co to the use of the Class. This attribute is initially False.

• If and *only* if the static attribute verbose is true, then the Class construc
produce an output.

• The Class must have a static method called `doc` that returns the docun
string. The content of the documentation must be read from a `Vector.do`

• Methods from the Vector Class should never modify the current instanc

• The Vector Class must have at least the following public methods and t
- `magnitude()`
- `normalize()`
- `add(Vector $rhs)`
- `sub(Vector $rhs)`
- `opposite()`
- `scalarProduct($k)`
- `dotProduct(Vector $rhs)`
- `cos(Vector $rhs)`
- `crossProduct(Vector $rhs)`

• Running the `main_02.php` script generates an output like the one in th

☐ Yes ☐

Ex03

The Matrix Class

• The Matrix Class must have seven Class constants: `IDENTITY`, `SCALE`, `TRANSLATION` and `PROJECTION`.

• The Class' constructor is expecting an array. The following keys are re
- `preset` : the matrix type to create, mandatory. The value must be one c
previoulsy defined.
- `scale` : the scale factor, mandatory when `preset` is `SCALE`.
- `angle` : the rotation angle in radians, mandatory when `preset` is `RX`,
- `vtc` : translation vector, mandatory when `preset` is `TRANSLATION`.
- `fov` : projection field of view in degrees, mandatory when `preset` is w
- `ratio` : projected image ratio, mandatory when `preset` is `PROJECTION`
- `near` : projection's near clipping plane, mandatory when `preset` is wo
- `far` : projection's far clipping plane, mandatory when `preset` is worth

• The Matrix Class must have a `__toString` method which output matcl

• The Class must include a Boolean static attribute called `verbose` to co
to the use of the Class. This attribute is initially False.

• If and *only* if the static attribute verbose is true, then the Class construc
produce an output.

• The Class must have a static method called `doc` that returns the docun
string. The content of the documentation must be read from a `Matrix.do`

• Methods from the Matrix Class should never modify the current instanc

• The Matrix Class must have at least the following public methods and t
- `mult(Matrix $rhs)`
- `transformVertex(Vertex $vtx)`

• Running the `main_03.php` script generates an output like the one in th

☐ Yes ☐

Ex04

The Camera Class

• The Class' constructor is expecting an array. The following keys are re
- `origin` : The vertex positioning the camera in the world coordinate syst
- `orientation` : Rotation matrix orienting the camera in the world coordi
- `width` : Width in pixel of the desired image. Is used to compute the rati
the `ratio` key.
- `height` : Height in pixel of the desired image. Is used to compute the ra
the `ratio` key.
- `ratio` : Image's ratio. Not compatible with the `width` and `height` key
- `fov` : The projected image's field of view in degrees.
- `near` : The near clipping plane.
- `far` : The far clipping plane.

• The Camera Class must have a `__toString` method which output mat

• The Class must include a Boolean static attribute called `verbose` to co
to the use of the Class. This attribute is initially False.

• If and *only* if the static attribute verbose is true, then the Class construc
produce an output.

• The Class must have a static method called `doc` that returns the docun
string. The content of the documentation must be read from a `Camera.do`

• The Camera Class must have at least the following public methods and
- `watchVertex(Vertex $worldVertex)`

• Running the `main_04.php` script generates an output like the one in th

☐ Yes ☐

Ex05 - Part 1

The Triangle Class

• The Class' constructor is expecting an array. The following keys are re
- `A` : Vertex of the first point of the triangle, mandatory.
- `B` : Vertex of the second point of the triangle, mandatory.
- `C` : Vertex of the third point of the triangle, mandatory.

• The Triangle Class must have a `__toString` method which output mat

• The Class must include a Boolean static attribute called `verbose` to co
to the use of the Class. This attribute is initially False.

• If and *only* if the static attribute verbose is true, then the Class construc
produce an output.

• The Class must have a static method called `doc` that returns the docun
string. The content of the documentation must be read from a `Triangle.`

• There are no mandatory methods for your Triangle Class. However, w

be useful. If the following methods aren't implemented it's not eliminatory to the bonus part. However, it's possible to give more points to reward re following methods, this exercise is worth 2 points:

- To be able to iterate or map the vertices of the triangle. ==> **+1 point**
- To be able to iterate or map the edges of the triangle. ==> **+1 point**
- To be able to sort the vertices or the edges under certain conditions. ==
- Any additional, functional and relevant features. ==> **+1 point** up to a

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

Ex05 - Part 2

The Render Class

• The Class' constructor is expecting an array. The following keys are re
- `width` : The generated image's width, mandatory.
- `height` : The generated image's height, mandatory.
- `filename` : Filename with which the created png image will be saved,

• The Render Class must have three Class constants: `VERTEX`, `EDGE` et F used to select the rendering mode.

• The Class must include a Boolean static attribute called `verbose` to co to the use of the Class. This attribute is initially False.

• If and *only* if the static attribute verbose is true, then the Class construc produce an output.

• The Class must have a static method called `doc` that returns the docun string. The content of the documentation must be read from a `Render.do`

• The Render Class must have at least the following public methods and
- `renderVertex(Vertex $screenVertex)`
- `renderTriangle(Triangle $triangle, $mode)`
- `develop()`

• The VERTEX mode works ==> **+1 point**
• The EDGE mode works ==> **+1 point**
• The RASTER mode works ==> **+3 point**

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

Ex06 - Bonus

The Texture Class

Don't try to nitpick on this one. If the project is clean and has fully functic 10 points (not half with excuses or explanations).
We're no beasts.

You can discuss about it and go get a coffee. ☕
Congratulations.

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

Ratings

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

☐ Ok ☐ Outstanding

☐ Empty work ☐ Incomplete work W Invalid compilation

I Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation

Terms of use for video surveillance (https://signin.intra.42.fr/legal/terms/1) Rules of procedure (https://signin.intra.42.fr/legal/terms/4) Declaration on the use of cookies (https://signin.intra.42.fr/legal/terms/2) General term of us (https://signin.intra.42.f