

(https://profile.intrav2.42.fr)



Go to

Scale for project Piscine PHP (/projects/piscine-php) / D06 (/projects/piscine-php-d06)

All projects (/projects)



ory List projects

jects.intrav2.42.fr

vogsphere@vogsphere.42.fr:intra/2015/activities/d06/pguillot



Your projects



Guidelines

Libft (/projects/libft)

Get Next Line

Tout le monde est censé corriger et correcteur

Faites alors un git clone du repo

Commencez la notation

Si il n'y a aucun rendu, mettez 0 et arrêtez l'évaluation. Discutez de ce qui a pose probleme (pas forcément

au niveau technique) et évitez la meme situation demain.

forum.intrav2.42.fr

Il est nécessaire d'avoir posséder tous les points d'un exercices pour passer au suivant. Le premier exercice qui n'obtient pas points stoppe la notation.



Wolf3d

Est éliminatoire (pas d'évaluation de l'exercice, fin de la notation) :

neta.intrav2.42.fr

Utiliser des parties de PHP interdites, en particulier les notions objet du j07 et du j08.

- Avoir une sortie qui diffère sur le fond de la sortie attendue (information manquante, calcul faux, fonctionnalité manquante, etc).

- Avoir une méthode ou un attribut en public sans raison. Si le corrigé justifie ce choix mais que vous n'êtes pas convaincu, vous restez seul juge de l'élimination.

- Ne pas respecter l'une de ces consignes :

• Une seule et unique classe par fichier.

• Un fichier qui contient la définition d'une classe ne doit rien contenir d'autre, à part des require ou require_once quand c'est nécessaire.

• Un fichier contenant une classe doit TOUJOURS être nommé sous la forme ClassName.class.php.

• Une classe doit TOUJOURS être accompagnée d'un fichier de documentation dont le nom doit TOUJOURS être de la forme ClassName.doc.txt.

• La documentation d'une classe doit TOUJOURS être utile et correspondre à l'implémentation.

• Une classe doit TOUJOURS proposer un attribut statique boolean verbose.

• Une classe doit TOUJOURS proposer une méthode statique doc renvoyant le contenu du fichier de documentation sous forme d'une chaîne de caractères.

• Un dossier de rendu d'un exercice doit contenir les fichiers des exercices précédents, qu'ils soient identiques, altérés ou nouveaux.

N'est pas éliminatoire :

- Avoir une sortie qui diffère dans sa forme de la sortie attendue (un espace en plus ou en moins, etc).

- Avoir une méthode ou un attribut qui n'a pas exactement le bon nom mais qui garde tout de même la bonne sémantique.

- Résoudre un exercice avec un algorithme différent de celui proposé dans l'énoncé tant que le résultat est identique.


Ratings


Define the type of error (if there is an error), which ended the correction.


Ok

Empty work Incomplete work No author file Invalid compilation  Norme Cheat Crash

Attachments

 Subject (<https://cdn.intrav2.42.fr/pdf/pdf/43/1c998daee903b647c05e67c2e6d39ea0-j06.pdf>)

 Intra - Elearning - Subnotions (<https://elearning.intrav2.42.fr/notions/41/subnotions>)

 resources (/uploads/document/document/39/4ff4f1a80f5a87a89ec05d22f7158db4-resources.tar)

Sections

La classe Color

La classe Color

- La classe Color doit posséder trois attributs publics entiers red, green et blue qui vous serviront à représenter les composantes d'une couleur.
- Le constructeur de la classe attend un tableau. Une instance doit pouvoir être construite, soit en passant une valeur pour la clef 'rgb' qui sera décomposée en trois composantes rouge, verte et bleue, soit en passant une valeur pour les clefs 'red', 'green' et 'blue' qui représentent directement les trois composantes.
- Chacune des valeurs pour les quatre clefs possibles est convertie en entier avant utilisation.
- La classe Color doit proposer une méthode __toString dont la sortie correspond à la sortie attendue.
- La classe doit contenir un attribut verbose statique de type boolean permettant de contrôler les affichages liés à l'utilisation de la classe. Cet attribut est initialement à la valeur False.
- Si et seulement si l'attribut statique verbose est vrai, alors le constructeur et le destructeur de la classe produisent une sortie conforme à la sortie d'exemple.
- La classe propose une méthode statique doc retournant une documentation courte de la classe sous forme de chaîne de caractères. Le contenu de la documentation est lu depuis un fichier Color.doc.txt.
- La classe propose une méthode add permettant d'ajouter les composantes de l'instance courante aux composantes d'une instance passée en paramètre. La couleur résultat est une nouvelle instance.
- La classe propose une méthode sub permettant de soustraire les composantes d'une instance passée en paramètre aux composantes de l'instance courante. La couleur résultat est une nouvelle instance.
- La classe doit proposer une méthode mult permettant de multiplier les composantes de l'instance courante par un facteur passé en paramètre. La couleur résultat est une nouvelle instance.
- L'exécution du script main_00.php produit une sortie identique sur le fond au contenu du fichier main_00.out.

 Yes

No

La class Vertex

La class Vertex

- La classe Vertex possède des attributs privés pour représenter x, y, z, w et sa couleur.
- La couleur du vertex est toujours une instance de la classe Color.
- La classe Vertex fourni des accesseurs en lecture et en écriture pour ses cinq attributs.
- Le constructeur de la classe attend un tableau, et les clefs attendues sont les suivantes :
 - 'x' : l'abscisse du vertex, obligatoire.
 - 'y' : l'abscisse du vertex, obligatoire.
 - 'z' : l'abscisse du vertex, obligatoire.
 - 'w' : optionnel, vaut 1.0 par défaut.
 - 'color' : optionnel, vaut une instance fraiche de la couleur blanche par défaut.
- La classe contient un attribut verbose statique de type boolean permettant de contrôler les affichages liés à l'utilisation de la classe. Cet attribut est initialement à la valeur False.
- La classe Vertex propose une méthode `__toString` correspondant a la sortie attendue.
- Si et seulement si l'attribut statique verbose est vrai, alors le constructeur et le destructeur de la classe produisent une sortie conforme a la sortie d'exemple.
- La classe propose une méthode statique doc retournant une documentation courte de la classe sous forme de chaine de caractères. Le contenu de la documentation est lu depuis un fichier Vertex.doc.txt. Cette documentation est pertinente, utile et a jour.
- L'execution du script main_01.php produit une sortie identique sur le fond au contenu du fichier main_01.out.

 Yes

No

La classe Vector

La classe Vector

- La classe Vector doit posséder des attributs privés pour représenter les quatre caractéristiques x, y, z et w.
- La classe Vector doit fournir des accesseurs en lecture seule pour ses quatre attributs.
- Le constructeur de la classe attend un tableau et les clefs attendues sont les suivantes :
 - 'dest' : vertex de destination du vecteur, obligatoire.
 - 'orig' : vertex d'origine du vecteur, optionnel, vaut une instance fraiche du vertex x=0, y=0, z=0, w=1 par défaut.
- La classe contient un attribut verbose statique de type boolean permettant de contrôler les affichages liés à l'utilisation de la classe. Cet attribut est initialement à la valeur False.
- La classe Vector doit proposer une méthode `__toString` conforme a la sortie attendue.
- Si et seulement si l'attribut statique verbose est vrai, alors le constructeur et le destructeur de la classe produisent une sortie conforme a la sortie d'exemple.
- La classe propose une méthode statique doc retournant une documentation courte de la classe sous forme de chaine de caractères. Le contenu de la documentation est lu depuis un fichier Vector.doc.txt. Cette documentation

chaîne de caractères. Le contenu de la documentation est lu depuis un fichier Vector.doc.txt. Cette documentation est pertinente, utile et à jour.

- Une méthode de la classe Vector ne modifie jamais l'instance courante.
- La classe Vector propose au moins les méthodes publiques suivantes et ces méthodes fonctionnent correctement :
 - magnitude
 - normalize
 - add
 - sub
 - opposite
 - scalarProduct
 - dotProduct
 - cos
 - crossProduct
- L'exécution du script main_02.php produit une sortie identique sur le fond au contenu du fichier main_02.out.

 Yes

No

La classe Matrix

La classe Matrix

- La classe Matrix propose sept constantes de classe : IDENTITY, SCALE, RX, RY, RZ, TRANSLATION et PROJECTION.
- Le constructeur de la classe attend un tableau, et les clefs attendues sont les suivantes :
 - 'preset' : type de matrice à construire, obligatoire. La valeur doit être l'une des constantes de classe précédentes.
 - 'scale' : le facteur d'échelle, obligatoire quand 'preset' vaut SCALE.
 - 'angle' : angle de rotation en radians, obligatoire quand 'preset' vaut RX, RY ou RZ.
 - 'vtc' : vecteur de translation, obligatoire quand 'preset' vaut TRANSLATION.
 - 'fov' : champs de vision de la projection en degrés, obligatoire quand 'preset' vaut PROJECTION.
 - 'ratio' : ratio de l'image projetée, obligatoire quand 'preset' vaut PROJECTION.
 - 'near' : plan de clipping proche de la projection, obligatoire quand 'preset' vaut PROJECTION.
 - 'far' : plan de clipping éloigné de la projection, obligatoire quand 'preset' vaut PROJECTION.
- La classe contient un attribut verbose statique de type boolean permettant de contrôler les affichages liés à l'utilisation de la classe. Cet attribut est initialement à la valeur False.
- La classe Matrix propose une méthode __toString conforme à la sortie attendue.
- Si et seulement si l'attribut statique verbose est vrai, alors le constructeur et le destructeur de la classe produisent une sortie conforme à la sortie d'exemple.
- La classe propose une méthode statique doc retournant une documentation courte de la classe sous forme de chaîne de caractères. Le contenu de la documentation est lu depuis un fichier Matrix.doc.txt. Cette documentation est pertinente, utile et à jour.
- Une méthode de la classe Matrix ne modifie jamais l'instance courante.
- La classe Matrix propose au moins les méthodes publiques suivantes et ces méthodes fonctionnent correctement :
 - mult
 - transformVertex
- L'exécution du script main_03.php produit une sortie identique sur le fond au contenu du fichier main_03.out.

 Yes

No

La classe Camera

La classe Camera

- Le constructeur de la classe attend un tableau, et les clefs attendues sont les suivantes :
 - 'origin' : Vertex positionnant la caméra dans le repère monde.
 - 'orientation' : Matrice de rotation orientant la caméra dans le repère monde.
 - 'width' : Largeur en pixels de l'image voulue. Sert à calculer le ratio. Incompatible avec la clef 'ratio'.
 - 'height' : Hauteur en pixels de l'image voulue. Sert à calculer le ratio. Incompatible avec la clef 'ratio'.
 - 'ratio' : Ratio de l'image. Incompatible avec les clefs 'width' et 'height'.
 - 'fov' : Le champs de vision de l'image projetée en degrés.
 - 'near' : Le plan de clipping proche.
 - 'far' : Le plan de clipping éloigné.
- La classe contient un attribut verbose statique de type boolean permettant de contrôler les affichages liés à l'utilisation de la classe. Cet attribut est initialement à la valeur False.
- La classe Camera propose une méthode __toString conforme a la sortie attendue.
- Si et seulement si l'attribut statique verbose est vrai, alors le constructeur et le destructeur de la classe produisent une sortie conforme a la sortie d'exemple.
- La classe propose une méthode statique doc retournant une documentation courte de la classe sous forme de chaine de caractères. Le contenu de la documentation est lu depuis un fichier Camera.doc.txt. Cette documentation est pertinente, utile et a jour.
- La classe Camera propose au moins les méthodes publiques suivantes et ces méthodes fonctionnent correctement :
 - watchVertex
- L'execution du script main_04.php produit une sortie identique sur le fond au contenu du fichier main_04.out.

✓ Yes

No

La classe Triangle

La classe Triangle

- Le constructeur de la classe Triangle attend un tableau et les clefs attendues sont les suivantes :
 - 'A' : Vertex du premier point du triangle, obligatoire.
 - 'B' : Vertex du second point du triangle, obligatoire.
 - 'C' : Vertex du troisième point du triangle, obligatoire.
- La classe contient un attribut verbose statique de type boolean permettant de contrôler les affichages liés à l'utilisation de la classe. Cet attribut est initialement à la valeur False.
- La classe Triangle propose une méthode __toString conforme a la sortie attendue.
- Si et seulement si l'attribut statique verbose est vrai, alors le constructeur et le destructeur de la classe produisent une sortie conforme a la sortie d'exemple.
- La classe propose une méthode statique doc retournant une documentation courte de la classe sous forme de chaine de caractères. Le contenu de la documentation est lu depuis un fichier Triangle.doc.txt. Cette documentation est pertinente, utile et a jour.
- Il n'y pas de méthodes obligatoires pour la classe Triangle. Toutefois en écrire quelques unes est une bonne idee. Ne pas avoir implemente les methodes suivantes n'est pas eliminaire et n'interdit pas l'accès a l'exercice bonus

ne pas avoir implémenté les méthodes suivantes n'est pas éliminatoire et n'interdit pas l'accès à l'exercice bonus. Toutefois, on peut mettre des points en plus pour récompenser la réflexion. L'exercice sans ces méthodes vaut 2 points.

- Itérer et/ou mapper les vertices du triangle ==> +1 point
- Itérer et/ou mapper les arêtes du triangle ==> +1 point
- Trier les vertices et/ou les arêtes selon certaines conditions ==> +1 point
- Fonctionnalités pertinentes supplémentaire ==> +1 point jusqu'à un maximum de 2

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



La classe Render

La classe Render

- Le constructeur de la classe Render attend un tableau et les clefs attendues sont les suivantes :
 - 'width' : Largeur de l'image générée, obligatoire.
 - 'height' : Hauteur de l'image générée, obligatoire.
 - 'filename' : Nom du fichier dans lequel sera sauvegardée l'image png créée, obligatoire.
- La classe Render doit proposer trois constantes de classe VERTEX, EDGE et RASTERIZE qui serviront à choisir le mode de rendu.
- La classe contient un attribut verbose statique de type boolean permettant de contrôler les affichages liés à l'utilisation de la classe. Cet attribut est initialement à la valeur False.
- Si et seulement si l'attribut statique verbose est vrai, alors le constructeur et le destructeur de la classe produisent une sortie conforme à la sortie d'exemple.
- La classe propose une méthode statique doc retournant une documentation courte de la classe sous forme de chaîne de caractères. Le contenu de la documentation est lu depuis un fichier Render.doc.txt. Cette documentation est pertinente, utile et à jour.
- La classe Render propose au moins les méthodes publiques suivantes et ces méthodes fonctionnent correctement :
 - renderVertex
 - renderTriangle
 - develop
- Le mode VERTEX fonctionne ==> L'exercice vaut 1 points
- Le mode EDGE fonctionne ==> L'exercice vaut 1 points
- Le mode RASTER fonctionne ==> L'exercice vaut 3 points

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



Bonus - La classe Texture

Bonus - La classe Texture

Ne cherchons pas la quadra-section capillaire. Si le rendu a des textures qui fonctionnent vraiment et correctement, pas juste à moitié, donner 10 points.
On est pas des bêtes.

Profitez-en pour discuter et aller boire un café ensemble.
Toutes mes félicitations.

Conclusion

Leave a comment on this correction

*** (required) Comment**

Domage pour les parenthèses oubliées sur getBlue.

Finish correction