

[Retour à toutes les fiches d'évaluation](#)

**Points gagnés**

**0**

## cub3d

Vous devez évaluer **2** étudiants dans cette équipe

### Introduction

Veuillez suivre les règles ci-dessous :

- ✓ Restez poli, courtois, respectueux et constructif tout au long du processus d'évaluation. Le bien-être de la communauté en dépend.
- ✓ Identifier avec l'étudiant ou le groupe dont le travail est évalué les dysfonctionnements possibles de son projet. Prendre le temps de discuter et de débattre des problèmes qui ont pu être identifiés.
- ✓ Il faut tenir compte du fait qu'il peut y avoir des différences dans la façon dont vos pairs ont pu comprendre les instructions du projet et l'étendue de ses fonctionnalités. Gardez toujours l'esprit ouvert et notez-les aussi honnêtement que possible. La pédagogie n'est utile que si l'évaluation par les pairs est faite sérieusement.

### Lignes directrices




Veuillez suivre les directives ci-dessous :

- ✓ Notez uniquement le travail qui a été remis au dépôt Git de l'étudiant ou du groupe évalué.
- ✓ Vérifiez que le dépôt Git appartient bien à l'étudiant(e) est bien celui attendu. Vérifiez également que « git clone » vide.
- ✓ Vérifiez soigneusement qu'aucun alias malveillant n'a et vous faire évaluer quelque chose qui n'est pas le contenu du référentiel officiel.
- ✓ Afin d'éviter toute surprise et le cas échéant, revoir ensemble les éventuels scripts utilisés pour faciliter la notation (scripts de test ou d'automatisation).
- ✓ Si vous n'avez pas terminé le devoir que vous allez évaluer, vous devez lire l'intégralité du sujet avant de commencer le processus d'évaluation.
- ✓ Utilisez les indicateurs disponibles pour signaler un dépôt vide, un programme non fonctionnel, une erreur de norme, une tricherie, etc. Dans ces cas, le processus d'évaluation se termine et la note finale est de 0, ou -42 en cas de tricherie. Cependant, à l'exception de la tricherie, les étudiants sont fortement encouragés à revoir ensemble le travail qui a été rendu, afin d'identifier les éventuelles erreurs qui ne devraient pas être répétées à l'avenir.
- ✓ N'oubliez pas que pendant toute la durée de la soutenance, aucun segfault ou autre terminaison inattendue, prématurée ou incontrôlée du programme ne sera tolérée, sinon la note finale est de 0. Utilisez le drapeau approprié.
- ✓ Vous ne devriez jamais avoir à modifier un fichier, à l'exception du fichier de configuration s'il existe. Si vous souhaitez modifier un fichier, prenez le temps d'expliquer les raisons à l'étudiant évalué et assurez-vous que vous êtes tous les deux d'accord.
- ✓ Il faut également vérifier l'absence de fuites mémoire. Toute mémoire allouée sur le tas doit être correctement libérée avant la fin de l'exécution.
- ✓ Vous êtes autorisé à utiliser n'importe lequel des différents outils disponibles sur l'ordinateur, tels que leaks, valgrind ou e\_fence. En cas de fuites de mémoire, cochez la case appropriée.

**Points gagnés****0**

## Pièces jointes

Veuillez télécharger les pièces jointes ci-dessous :

-  [sujet.pdf](#)
-  [minilibx\\_opengl.tgz](#)
-  [minilibx\\_mms\\_20200219\\_beta.tgz](#)

**Points gagnés**

**0**

## Partie obligatoire

### Nom de l'exécutable

Nom de l'exécutable

Vérifiez que le projet compile bien (sans re-link) lorsque vous exécutez la make commande et que le nom de l'exécutable est cub3D . Si ce n'est pas le cas, utilisez l'indicateur de compilation invalide à la fin de l'échelle.

Oui

Non

### Fichier de configuration

Vérifiez que vous pouvez configurer TOUS les éléments suivants dans le fichier de configuration. Le formatage doit être conforme à la description du sujet.

chemin de texture nord ☒ NON

chemin de texture nord ☒ NON

chemin de texture est ☒ EA

chemin de texture sud ☒ DONC

chemin de texture ouest ☒ NOUS

couleur du sol ☒ F

couleur du plafond ☒ C

**Points gagnés**

**0**

la carte (voir sujet pour les détails de configuration de la carte)

Vérifiez également que le programme renvoie une erreur et se ferme correctement lorsque le fichier de configuration est mal configuré (par exemple une clé inconnue, des clés doubles, un chemin invalide...) ou si le nom du fichier ne se termine pas par l' .cub extension. Si ce n'est pas le cas, la défense est terminée et utilisez le flag approprié travail incomplet, crash....

Oui

Non

## Éléments techniques de l'affichage

Nous allons évaluer les éléments techniques de l'affichage. Exécutez le programme et effectuez les tests suivants. Si au moins un échoue, aucun point ne sera attribué pour cette section. Passez à la suivante.

Une fenêtre doit s'ouvrir au lancement du programme. Une fenêtre doit s'ouvrir au lancement du programme.

Une image représentant l'intérieur d'un labyrinthe doit être affichée à l'intérieur de la fenêtre.

Masquez tout ou partie de la fenêtre soit en utilisant une autre fenêtre, soit en utilisant les bordures de l'écran, puis réduisez la fenêtre et agrandissez-la à nouveau. Dans tous les cas, le contenu de la fenêtre doit rester cohérent.

Oui

Non

## Événements de base de l'utilisateur

### Points gagnés

0

Dans cette section, nous allons évaluer les événements du programme. Exécutez les 3 tests suivants. Si au moins un échoue, cela signifie qu'aucun point ne sera attribué pour cette section. Passez à la suivante.

Cliquez sur la croix rouge en haut à gauche de la fenêtre. La fenêtre doit se fermer et le programme doit se terminer proprement.

Appuyez sur la touche ESC. La fenêtre doit se fermer et le programme doit sortir proprement. Dans le cas de ce test, nous accepterons qu'une autre touche sorte du programme, par exemple, Q.

Appuyez sur les quatre touches de déplacement (nous acceptons les touches WASD ou ZQSD) dans l'ordre de votre choix. Chaque pression de touche doit générer un résultat visible sur la fenêtre, comme le déplacement/la rotation d'un joueur.

Oui

Non

## Mouvements

Dans cette section, nous allons évaluer l'implémentation du mouvement/orientation du joueur à l'intérieur du labyrinthe. Exécutez les 5 tests suivants. Si au moins un échoue, cela signifie qu'aucun point ne sera attribué pour cette section.

L'orientation d'apparition du joueur sur la première image doit être conforme au fichier de configuration, testez pour chaque orientation cardinale (N, S, E, W).

Appuyez sur la flèche gauche puis sur la flèche droite. La vue du joueur doit

pivoter vers la gauche puis vers la droite comme si la tête du joueur bougeait.

Appuyez sur W (ou Z) puis S. La vue du joueur doit avancer puis reculer en ligne droite.

Appuyez sur A (ou Q) puis sur D.

La vue du joueur doit aller vers la gauche puis vers la droite.

Pendant ces quatre mouvements, l'affichage était-il fluide ? Par fluide, nous entendons si le jeu est « jouable » ou bien est-il lent.

**Points gagnés**

**0**

Oui

Non

## Murs

Dans cette section, nous allons évaluer les murs du labyrinthe. Exécutez les 4 tests suivants. Si au moins un échoue, cela signifie qu'aucun point ne sera attribué pour cette section.

La texture du mur varie en fonction de l'orientation du mur (nord, sud, est, ouest). Vérifiez que les textures des murs et la perspective sont visibles et correctes.

Vérifiez que si vous modifiez le chemin d'une texture de mur dans le fichier de configuration, cela modifie la texture rendue lorsque le programme est réexécuté. Vérifiez également que si vous définissez un chemin inexistant, cela génère une erreur.

Vérifiez que les couleurs du sol et du plafond sont bien gérées lorsque vous les modifiez dans le fichier de configuration.

Oui

Non

## Gestion des erreurs

Dans cette section, nous allons évaluer la gestion des erreurs de votre programme. Exécutez les 4 tests suivants. Si au moins un test échoue, aucun point ne sera attribué pour cette section. Pas de points gagnés.

Points gagnés

0

Exécutez le programme en utilisant de nombreux arguments aléatoires. Même si le programme ne nécessite aucun argument, il est essentiel que ces arguments n'alternent pas ou ne créent pas d'erreurs non gérées.

Vérifiez qu'il n'y a pas de fuites de mémoire. Vous pouvez utiliser la commande `top` ou `leaks` dans un autre shell pour vérifier que l'utilisation de la mémoire est stable. La mémoire utilisée ne doit pas augmenter à chaque fois qu'une action est effectuée.

Faites rouler votre bras ou votre visage sur le clavier. Le programme ne doit pas présenter de comportements étranges et doit rester fonctionnel.

Modifier la carte. Le programme ne doit pas présenter de comportements étranges et doit rester fonctionnel si la carte est bien configurée, sinon il doit générer une erreur.

Oui

Non

## Partie bonus

### Quand je serai plus vieux, je serai John Carmack

Nous examinerons vos bonus si et seulement si votre partie obligatoire est excellente. Cela signifie que vous devez terminer la partie obligatoire du début à la fin et que votre gestion des erreurs doit être irréprochable, même en cas

d'utilisation tordue ou mauvaise. Ainsi, si la partie obligatoire n'a pas marqué tous les points lors de cette soutenance, les bonus seront totalement ignorés.

Regardez la partie bonus du sujet et ajoutez un point par œuvre et entièrement fonctionnel.

**Points gagnés****0**

Notez-le de 0 (échec) à 5 (excellence)



0

1

2

3

4

5

## Notes

 D'ACCORD Remarquable Travail vide Travail incomplet Compilation invalide Normes Tricher Accident Concernant les situations Fuites Fonctions interdites



© 2024 42evals. Tous droits réservés

**Points gagnés**

**0**