

Commencé le	mardi 22 février 2022, 09:01
État	Terminé
Terminé le	mardi 22 février 2022, 11:29
Temps mis	2 heures 28 min
Note	31,00 sur 35,00 (89%)

Description

L'examen comporte 13 questions (les deux dernières sont séparés en huit et quatre sous-question) pour un total de 35points.

L'examen est **individuel** et toute documentation est permise.

Vous n'avez pas besoin de prendre de note écrite, l'intra se remplit entièrement sur moodlequiz.

Assurez vous de bien lire les énoncés. Si vous jugez qu'il manque des informations pour répondre adéquatement à une question, énoncez les suppositions que vous avez faites pour y répondre.

Bonne chance et bon courage!

Question 1

Terminer

Note de 2,00 sur 2,00

Expliquez en vos mots la différence entre une primitive géométrique et un contexte graphique.

Donnez un exemple concret pour chacun.

Une primitive géométrique est une forme (par exemple un triangle, un rectangle, un segment, ...) tandis que le contexte graphique englobe tous les caractéristiques de cette forme (couleur, texture, épaisseur et type de trait, type de remplissage ...)

Commentaire :

Question 2

Terminer

Note de 1,00 sur 1,00

Vous voulez simuler un effet de fumée simple dans votre scène en appliquant une image translucide.

Quel est le nom de l'opération que vous devez effectuer sur les pixels afin d'obtenir cet effet?

On utilisera un nuanceur de fragments pour que les pixels aient une fusion des couleurs des différents objets (ex: fusion entre la couleur d'un mur et la couleur de la vapeur d'eau).

Dans le code, cela se traduira par un appel à `glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_COLOR)` ou `glBlendFunc(GL_SRC_ALPHA, GL_ONE)` en fonction de l'effet désiré (ou avec un autre mode).

Commentaire :

Question 3

Terminer

Note de 1,00 sur 1,00

Une translation peut être représentée comme une addition tandis que les autres transformations sont des multiplications.

Comment est-il possible d'avoir une seule matrice de transformations qui combine des multiplications et des additions?

On utilise une matrice 4x4. On utilise la sous-matrice 3x3 (avec a,b,c,e,f,g,i,j,k) pour le scale et rotate, tandis qu'on utilise la sous-matrice 3x1 (avec d,h,l) pour le translate.

[a, b, c, d]

[e, f, g, h]

[i, j, k, l]

[0, 0, 0, m]

On multiplie ensuite cette matrice M par la matrice des coordonnées A (x, y, z, 1) pour obtenir A' avec les transformations voulues.

Commentaire :

Question 4

Terminer

Note de 1,00 sur 1,00

Expliquez ce que représente un point de fuite.

Le point de fuite est, dans une image 2D, le point vers lequel convergent différentes droites. Ces droites sont en réalité parallèles. C'est grâce à cela qu'on a un effet de profondeur.

Il est utilisé pour représenter la perspective (à l'opposé d'une projection orthographique).

Commentaire :

Vous auriez pu le formuler ainsi: Un point virtuel vers lequel converge des droites 3D lorsqu'elles sont projetées en 2D

Question 5

Terminer

Note de 3,00 sur 3,00

Énumérez les différents types de sources lumineuses vus en cours et donnez un exemple concret de chacune.

On a vu 4 types de sources lumineuses :

- positionnelle (ex: une lampe au plafond d'une pièce)
- directionnelle (ex: les rayons du soleil qui traversent des nuages)
- spot (ex: les phares d'une voiture)
- lumière ambiante (ex: ionisation de l'air par des radiations gamma suite à une explosion d'un réacteur nucléaire)

Commentaire :

"ionisation de l'air par des radiations gamma suite à une explosion d'un réacteur nucléaire" Quelqu'un a écouté Chernobyl récemment...

Question **6**

Terminer

Note de 1,00 sur 1,00

On parle de face avant ou de face arrière dans OpenGL (GL_FRONT et GL_BACK) quelle convention est utilisée pour différencier ces faces?

Par convention, la face avant est celle où l'ordre des sommets est dans le sens anti-horaire.

Commentaire :

Par convention, la face avant est celle où l'ordre des sommets est dans le sens anti-horaire dans la projection 2D

Question **7**

Terminer

Note de 1,00 sur 1,00

Dans le passé, certains systèmes utilisaient des quadrilatères plutôt que des triangles comme primitive fondamentale. Donnez la raison pour laquelle, selon vous, les triangles sont un meilleur choix que les quadrilatères et pourquoi on n'utilise plus les quadrilatères comme primitive de base. Justifiez votre réponse en utilisant des termes ou notions en lien avec le cours et l'infographie 3D.

Les triangles sont plus permissifs pour créer d'autres formes.

De plus, utiliser seulement des triangles permet de traiter seulement une seule forme, donc un seul format d'informations. Cela simplifie le transfert d'informations entre CPU et GPU et les signatures des méthodes pour les bibliothèques comme OpenGL

Commentaire :

Question 8

Terminer

Note de 0,50 sur 2,00

Une sphère est tracée dans une application graphique 3D. Cette sphère est entièrement contenue dans un volume de visualisation qui respecte le rapport d'aspect et elle est entièrement visible dans la fenêtre à l'écran.

a) Si cette application utilise une projection orthographique, est-ce que la silhouette de la sphère à l'écran sera toujours un cercle, quelle que soit sa position à l'écran ? Justifiez

b) Si cette application utilise une projection perspective, est-ce que la silhouette de la sphère à l'écran sera toujours un cercle, quelle que soit sa position à l'écran ? Justifiez.

a) La silhouette sera toujours une sphère. Les sommets sont projetés parallèlement vers le plan de projection, donc la forme projetée à l'écran en 2D sera un cercle de même rayon que la sphère.

b) La silhouette sera toujours une sphère. Les sommets sont projetés vers l'oeil de la caméra. On aura toujours une forme de cercle projetée à l'écran en 2D. Cependant le rayon sera plus petit que le rayon réel de la sphère, en fonction de la distance entre la sphère et la caméra.

Remarque: c'est comme cela que l'humain perçoit les choses. Une sphère nous semble bien sphérique quand elle apparaît dans notre champ de vision.

Commentaire :

a) Puisque le rapport d'aspect est respecté, oui elle sera toujours celle d'un cercle.

b) Non, si la sphère n'est pas centrée avec l'axe de vue de la projection perspective la silhouette n'est pas nécessairement celle d'un cercle

Question 9

Terminer

Note de 1,00 sur 1,00

Décrivez (en vos propres mots) comment est utilisé le tampon de profondeur pour la suppression des parties cachées d'un objet graphique. Prenez soin de bien identifier les attributs des fragments ou les valeurs dans les tampons qui sont utilisés. Dites aussi ce que contient ou peut contenir ce tampon de profondeur.

Pour utiliser un test de profondeur, il faut d'abord l'activer avec `glEnable(GL_DEPTH_TEST);`

On compare pour chaque fragment sa profondeur `z` (`gl_FragCoord.x`) avec la valeur contenue dans le tampon grâce à l'appel de `glDepthFunc(GLenum func);`

`GLenum func` correspond à la fonction de comparaison à utiliser. (`GL_NEVER`, `GL_LESS`, `GL_EQUAL`, `GL_LEQUAL`, `GL_GREATER`, `GL_NOTEQUAL`, `GL_GEQUAL` ou `GL_ALWAYS`)

Remarque: par défaut, la fonction de comparaison utilisée est `GL_LESS`.

Ainsi, chaque fragment qui passe le test sera retenu dans le tampon et pourra ensuite être affiché à l'écran

Commentaire :

Attention la profondeur du fragment est `gl_FragCoord.z` ... `x,y` sont les positions à l'écran du fragment.

Question **10**

Terminer

Note de 2,00 sur 2,00

Classez les différents modèles d'illumination vus en cours selon la complexité des calculs nécessaires (Du plus léger au plus intensif).

Du plus léger au plus intensif:

- Modèle de Lambert
- Modèle de Gouraud
- Modèle de Phong

Commentaire :

Question **11**

Terminer

Note de 2,00 sur 2,00

Dans le TP1, on vous a demandé d'utiliser des Vertex Buffer Object (VBO) pour afficher la théière bien connue. Visuellement, on ne constate aucune différence si on utilise des VBO ou si on utilise la paire « glBegin / glEnd ».

a) Expliquez l'intérêt d'utiliser des VBO dans un logiciel d'infographie. Expliquez ce qui se passe lorsqu'on en utilise (par rapport à ne pas en utiliser).

b) Pourquoi ne remarque-t-on aucune différence entre l'utilisation ou non de VBO dans ce TP1 ? (La théière peut s'afficher dans les deux cas.)

a) les VBO réduisent la quantité d'information à transmettre entre le CPU et la GPU. Ceci offre donc une meilleure performance.

Au lieu de transmettre les 3 sommets pour chacun des triangles (et donc faire une redondance d'information pour les sommets communs et de nombreux envois), on préfère transmettre un tableau contenant tous les sommets (pas de redondance) et un tableau de connectivité reliant les sommets. Ainsi on transmet beaucoup moins de coordonnées (floats).

b) Dans le TP1, on affichait seulement une théière. Le CPU et GPU peuvent facilement gérer cela car il n'y a pas beaucoup d'informations à transmettre.

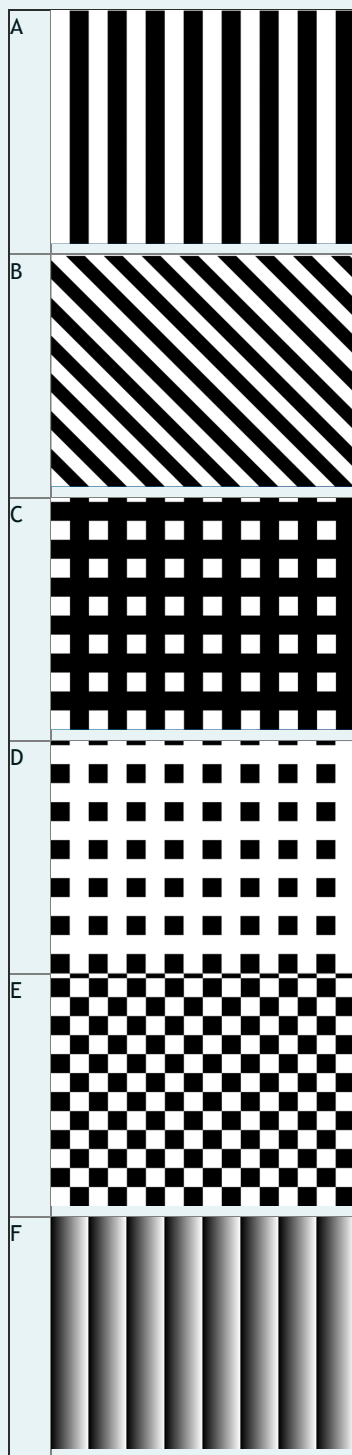
Cependant, dans un gros jeu vidéo, il devient nécessaire d'utiliser des VBO sinon il risque d'y avoir des problèmes de performances pour l'affichage graphique.

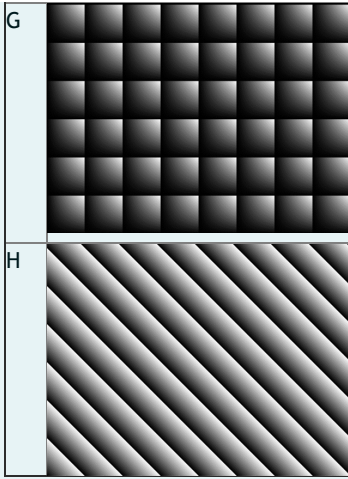
Commentaire :

Fragments [10 points]

Vous devrez, dans les questions suivantes, associer les 8 images ci-dessous au nuanceur de fragments les ayant générées. Vous devrez aussi écrire le code des 2 nuanceurs manquants.

Pour les associations inscrivez seulement "Figure {lettre de la figure}" ou la **lettre de la figure seule**. (La casse n'est pas importante)





Question 12

Terminer

Note de 1,00 sur 1,00

NuanceurFragments1

```
void main(void)
{
    if (mod(gl_FragCoord.x, 100.0) < 50.0)
        coul = vec4(1.0, 1.0, 1.0, 1.0);
    else
        coul = vec4(0.0, 0.0, 0.0, 0.0);
}
```

Réponse :

A

Question 13

Terminer

Note de 1,00 sur 1,00

NuanceurFragments2

```
void main(void)
{
    float c = mod(gl_FragCoord.x, 100.0) / 100;
    coul = vec4(c, c, c, 1.0);
}
```

Réponse :

F

Question 14

Terminer

Note de 1,00 sur 1,00

NuanceurFragments3

```
void main(void)
{
    if (mod(gl_FragCoord.x, 100.0) < 50.0 && mod(gl_FragCoord.y, 100.0) < 50.0)
        coul = vec4(1.0, 1.0, 1.0, 1.0);
    else
        coul = vec4(0.0, 0.0, 0.0, 0.0);
}
```

Réponse :

C

Question 15

Terminer

Note de 1,00 sur 1,00

NuanceurFragments4

```
void main(void)
{
    if (mod(gl_FragCoord.x, 100.0) < 50.0 && mod(gl_FragCoord.y, 100.0) < 50.0
        || mod(gl_FragCoord.x, 100.0) > 50.0 && mod(gl_FragCoord.y, 100.0) > 50.0)
        coul = vec4(1.0, 1.0, 1.0, 1.0);
    else
        coul = vec4(0.0, 0.0, 0.0, 0.0);
}
```

Réponse :

E

Question 16

Terminer

Note de 0,00 sur 1,00

NuanceurFragments5

```
void main(void)
{
    float c = (mod(gl_FragCoord.x, 100.0) / 100) * (mod(gl_FragCoord.y, 100.0) / 100);
    color = vec4(c, c, c, 1.0);
}
```

Réponse :

H

Question 17

Terminer

Note de 1,00 sur 1,00

NuanceurFragments6

```
void main(void)
{
    if (mod(gl_FragCoord.x + gl_FragCoord.y, 100.0) < 50.0)
        color = vec4(1.0, 1.0, 1.0, 1.0);
    else
        color = vec4(0.0, 0.0, 0.0, 1.0);
}
```

Réponse :

Question 18

Terminer

Note de 2,00 sur 2,00

Écrire dans l'espace le code du premier nuanceur manquant.

Vous n'avez pas à écrire ce qu'il reçoit en entrée, le #version ou tout autre information, simplement ce que le main() devrait contenir comme les six autres nuanceurs.

```
// figure D
void main(void) {
    if (mod(gl_FragCoord.x, 100.0) > 50.0 || mod(gl_FragCoord.y, 100.0) > 50.0)
        coul = vec4(1.0, 1.0, 1.0, 1.0);
    else
        coul = vec4(0.0, 0.0, 0.0, 0.0);
}
```

Commentaire :

Question 19

Terminer

Note de 1,00 sur 2,00

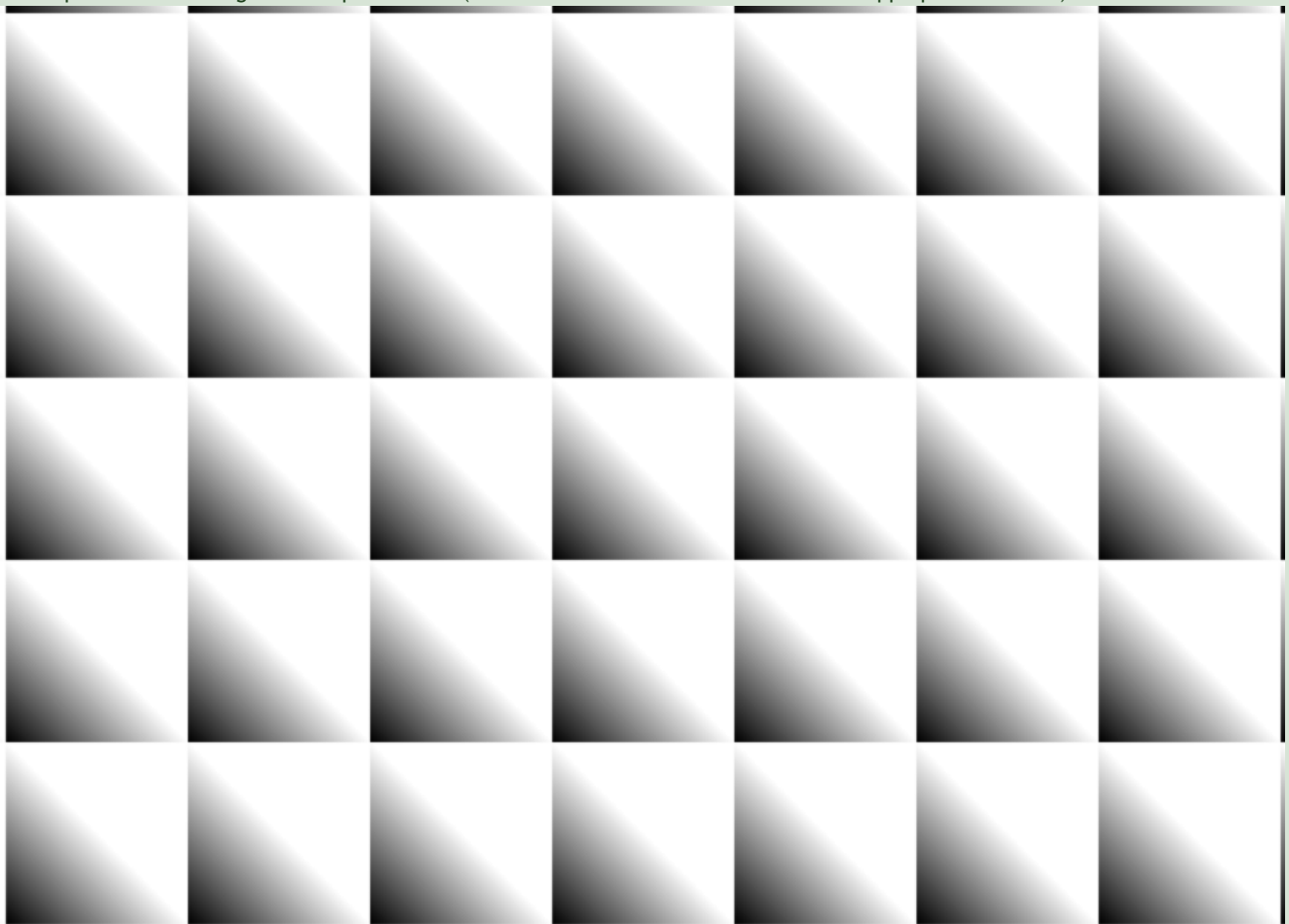
Écrire dans l'espace le code du second nuanceur manquant.

Vous n'avez pas à écrire ce qu'il reçoit en entrée, le #version ou tout autre information, simplement ce que le main() devrait contenir comme les six autres nuanceurs.

```
// figure G
void main(void) {
    float c = (mod(gl_FragCoord.x, 100.0) / 100) + (mod(gl_FragCoord.y, 100.0) / 100);
    color = vec4(c, c, c, 1.0);
}
```

Commentaire :

le code que vous avez produit la figure G mais trop éclairé. Le code pour G était fourni. Avec une petite modification votre nuanceur aurait produit une des figures manquantes la H (en additionnant les coordonnées avant d'appliquer le modulo)



Stencils [8 points]

Pour les attributs de fragments suivants, vous décidez de tester manuellement la valeur qui sera inscrite dans les tampons à l'écran. Pour chacun des attributs, peu importe le résultat des tests, donnez la valeur de chacun des tampons et utiliser ces nouvelles valeurs pour le fragment suivant. Si un fragment ne **réussit pas** les tests, spécifiez le premier test qui va échouer et la/les raison(s).

Aussi, les valeurs des tampons après les tests sur le Fragment1 seront utilisées pour les tests avec le Fragment2 et ainsi de suite.

Les lignes de code fournis détaillent les seules opérations sur les fragments (si un test **n'y figure pas**, considérez qu'il est désactivé).

- a. FragColor1 = vec4 (1.0, 0.5, 0.5, 1.0) gl_FragCoord1 = vec4 (2.5, 2.5, 1.0, 1.0)
- b. FragColor2 = vec4 (0.3, 0.3, 0.3, 0.66) gl_FragCoord2 = vec4 (-17, 12, 0.3, 1.0)
- c. FragColor3 = vec4 (0.2, 0.75, 0.2, 0.35) gl_FragCoord3 = vec4 (15, 15, -10, 1.0)
- d. FragColor4 = vec4 (0.35, 0.35, 0.55, 0.8) gl_FragCoord4 = vec4 (4.4, 4.5, 4.6, 1.0)

Valeurs dans les tampons au début de votre analyse :

- e. Tampon de... profondeur = 1.0, stencil = 4, couleur = vec4 (0.8, 0.7, 0.0, 0.55) initial

Question 20

Terminer

Note de 1,50 sur 2,00

Indiquer les valeurs dans les tampons après la première série d'opérations

```
glEnable(GL_STENCIL_TEST);
glStencilFunc(GL_GREATER, 3, 2);
glStencilOp(GL_INCR, GL_INVERT, GL_KEEP);
glEnable(GL_DEPTH_TEST);
glDepthFunc(GL_LESS);
glEnable(GL_BLEND);
glBlendFunc(GL_ONE_MINUS_DST_COLOR, GL_ONE_MINUS_SRC_ALPHA);
```

profondeur = 1.0
 stencil = -4
 couleur = vec4(0.8, 0.7, 0.0, 0.55)

Commentaire :

Vous n'avez pas spécifier que le test de profondeur échoue (votre stencil a subi un GL_INVERT donc vous dites que le test de stencil réussi mais pas celui de profondeur)

Question **21**

Terminer

Note de 2,00 sur 2,00

Indiquer les valeurs dans les tampons après la deuxième série d'opération

```
glStencilFunc(GL_ALWAYS, 1, 1);  
glStencilOp(GL_KEEP, GL_KEEP, GL_REPLACE);  
glEnable(GL_DEPTH_TEST);  
glDepthFunc(GL_LEQUAL);  
glDisable(GL_BLEND);
```

profondeur = 0.3
stencil = -4
couleur = vec4(0.3, 0.3, 0.3, 0.66)

Commentaire :

Excellent, c'est ce qu'il fallait faire

Question **22**

Terminer

Note de 2,00 sur 2,00

Indiquer les valeurs dans les tampons après la troisième série d'opération

```
glDisable(GL_STENCIL_TEST);  
glStencilFunc(GL_ALWAYS, 4, 7);  
glStencilOp(GL_REPLACE, GL_KEEP, GL_REPLACE);  
glEnable(GL_DEPTH_TEST);  
glDepthFunc(GL_NOTEQUAL);  
glEnable(GL_BLEND);  
glBlendFunc(GL_ONE, GL_ONE);
```

profondeur = -10
stencil = -4
couleur = vec4(0.5, 1.0, 0.5, 1.0)

Commentaire :

Question **23**

Terminer

Note de 2,00 sur 2,00

Indiquer les valeurs dans les tampons après la quatrième série d'opération

```
glDisable(GL_STENCIL_TEST);  
glStencilFunc(GL_LEQUAL, 2, 2);  
glStencilOp(GL_INCR, GL_INCR, GL_REPLACE);  
glEnable(GL_DEPTH_TEST);  
glDepthFunc(GL_NOTEQUAL);  
glBlendFunc(GL_ONE, GL_ONE);
```

profondeur = 4.6
stencil = -4
couleur = vec4(0.35, 0.35, 0.55, 0.8)

Commentaire :
Excellent travail

Description

Assurez-vous d'avoir bien rempli les questions avant de faire la remise de votre examen.

[◀ Annonces](#)

Aller à...