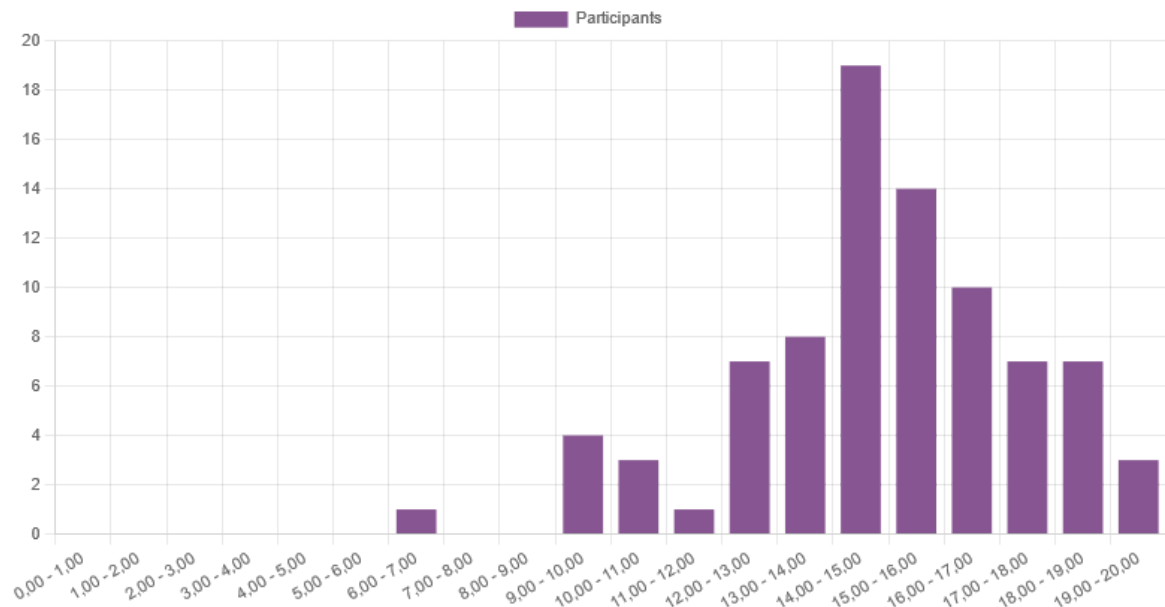


Commencé le	lundi 22 avril 2024, 09:30
État	Terminé
Terminé le	lundi 22 avril 2024, 11:32
Temps mis	2 heures 1 min
Note	15,29 sur 20,00 (76,46%)
Feedback	Moyenne de la classe : 14.90/20
	Médiane de la classe : 14.92/20
	Distribution des notes :



Description

INF2705 : Infographie (Hiver 2022)

Examen final

- L'examen dure 2h30 et est sur 20 points.
- Assurez-vous d'enregistrer vos réponses tout au long de l'examen pour éviter les problèmes d'enregistrement des réponses.
- Vous avez accès aux diapositives du cours sur MoodleExamen. Aucune autre documentation permise.
- Aucune calculatrice permise (vous avez la calculatrice Windows et une console Python sur votre poste).
- Vous pouvez utiliser une feuille brouillon.
- En cas de problème technique majeur, veuillez contacter le chargé de cours en communiquant avec les surveillants.
- Il n'y a pas de partie pratique, mais vous avez sur votre session une solution Visual Studio dans laquelle vous pouvez écrire du code. C'est essentiellement une feuille brouillon programmable et ne sera pas lue par le correcteur.

Question 1

Partiellement correct

Note de 1,42 sur 1,50

Pour chacun des éléments ci-dessous dans une scène avec un objet éclairé, indiquez s'il influence les calculs d'intensité pour chacune des composantes de l'illumination : la lumière émise par le matériau de l'objet, la réflexion ambiante, la réflexion diffuse et la réflexion spéculaire.

Élément**Éclairage(s) affecté(s)**

☐ émission ☐ réfl. amb. ☒ réfl. diff. ✓ ☒ réfl. spéc. ✓

Note de 1,00 sur 1,00

Position de la source lumineuse

La réponse correcte est :

- réfl. diff.
- réfl. spéc.

☐ émission ☐ réfl. amb. ☒ réfl. diff. ✓ ☒ réfl. spéc. ✓

Note de 1,00 sur 1,00

Normale de la surface

La réponse correcte est :

- réfl. diff.
- réfl. spéc.

☐ émission ☐ réfl. amb. ☐ réfl. diff. ☒ réfl. spéc. ✓

Note de 1,00 sur 1,00

Position de l'observateur

La réponse correcte est :

- réfl. spéc.

☐ émission ☐ réfl. amb. ☐ réfl. diff. ☒ réfl. spéc. ✓

Note de 1,00 sur 1,00

Brillance (*shininess*) du matériau

La réponse correcte est :

- réfl. spéc.

☒ émission ✓ ☒ réfl. amb. ✓ ☒ réfl. diff. ✓ ☒ réfl. spéc. ✓

Note de 1,00 sur 1,00

La réponse correcte est :

Couleurs du matériau

- émission
- réfl. amb.
- réfl. diff.
- réfl. spéc.

☒ émission ✗ ☒ réfl. amb. ☒ ☒ réfl. diff. ☒ ☒ réfl. spéc. ☒

Note de 0,67 sur 1,00

La réponse correcte est :

Couleurs de la source lumineuse

- réfl. amb.
- réfl. diff.
- réfl. spéc.

Question 2

Partiellement correct

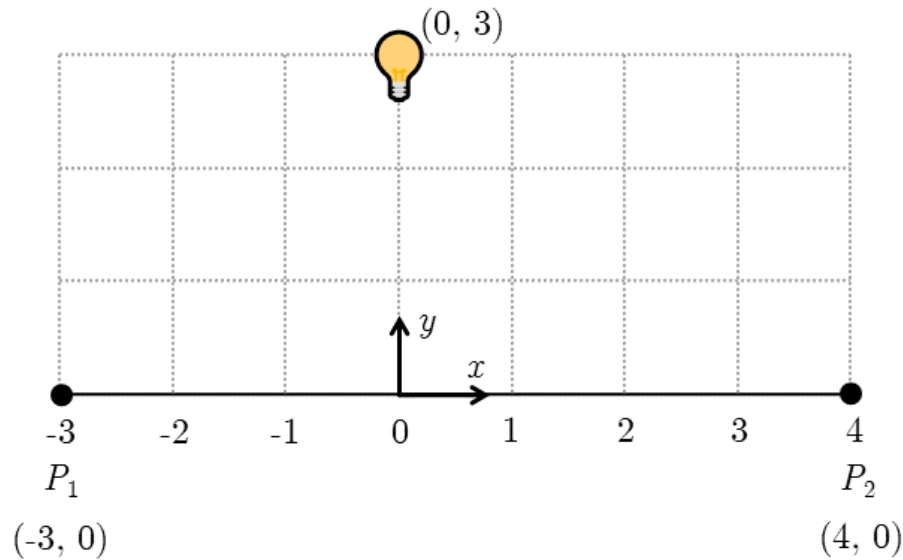
Note de 0,83 sur 1,00

Les modèles d'illumination de Lambert (*flat*), Gouraud et Phong sont des approches différentes pour calculer et interpoler l'intensité lumineuse sur la surface d'un objet dans une scène éclairée. Spécifiez, pour chacun des modèles, dans quelle étape du pipeline programmable sont effectués les calculs géométriques et les calculs d'intensité lumineuse.

Modèle	Calculs géométriques (<i>L</i> , <i>N</i> et <i>O</i>)	Calculs d'intensités (les couleurs réfléchies)
Lambert	Nuanceur de géométrie ✓	Nuanceur de géométrie ✓
Gouraud	Nuanceur de sommets ✓	Nuanceur de sommets ✓
Phong	Nuanceur de fragments ✗	Nuanceur de fragments ✓

Description

Soit la scène 2D ($z=0$ partout) suivante composée d'un segment de droite entre deux sommets P_1 et P_2 et d'une source lumineuse. On souhaite comparer le comportement de la **réflexion diffuse** sur ce segment selon les modèles d'illumination de Gouraud et de Phong.



On suppose que :

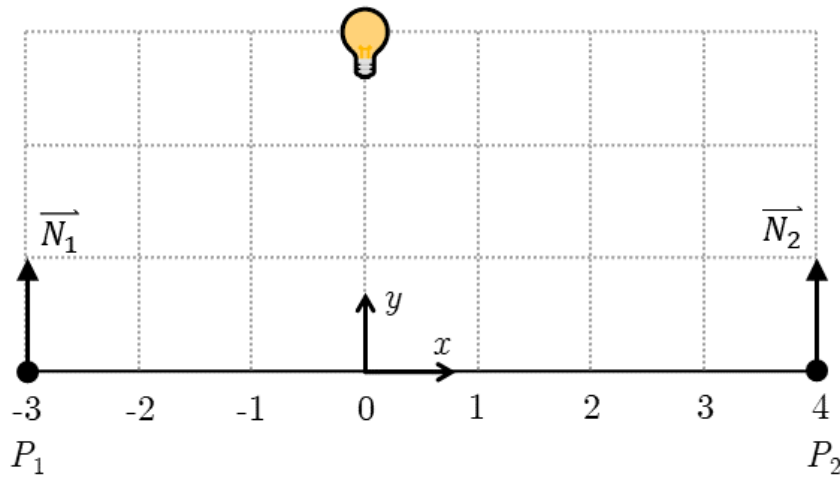
- la source de lumière est positionnelle
- il n'y a aucun sommet intermédiaire entre P_1 et P_2
- il n'y a que de la réflexion diffuse (pas ambiante ou spéculaire) et pas d'émission du matériau
- la source de lumière et le matériau du segment sont tous deux parfaitement blancs (intensité maximale)
- il n'y a aucune atténuation selon la distance.

Question 3

Partiellement correct

Note de 1,00 sur 2,00

Avec les normales suivantes :



Quelles sont les valeurs d'illuminations à P_1 ($x = -3$), P_2 ($x = 4$) et à $x = 0$? Répondez avec deux décimales.

Selon le modèle de Gouraud à P_1 : ✓ , à P_2 : ✗ et à $x = 0$: ✗

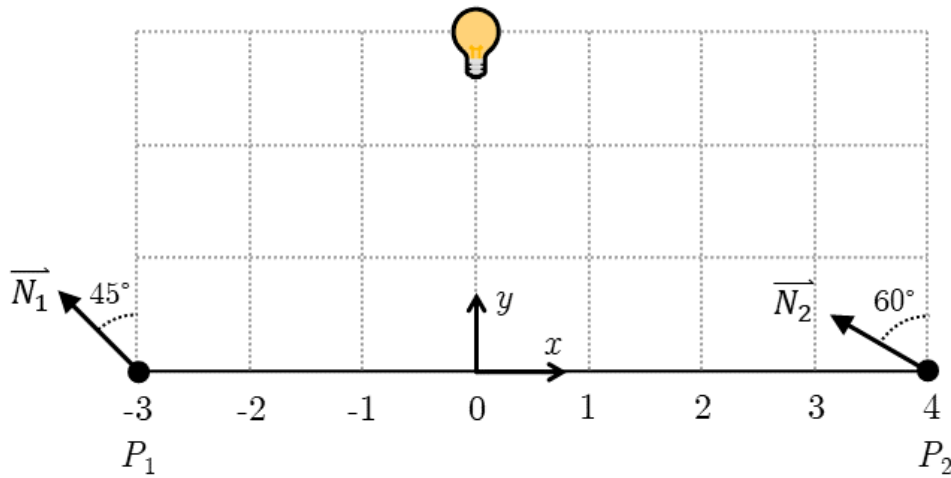
Selon le modèle de Phong à P_1 : ✓ , à P_2 : ✗ et à $x = 0$: ✓

Question 4

Partiellement correct

Note de 1,33 sur 2,00

Avec les normales suivantes :



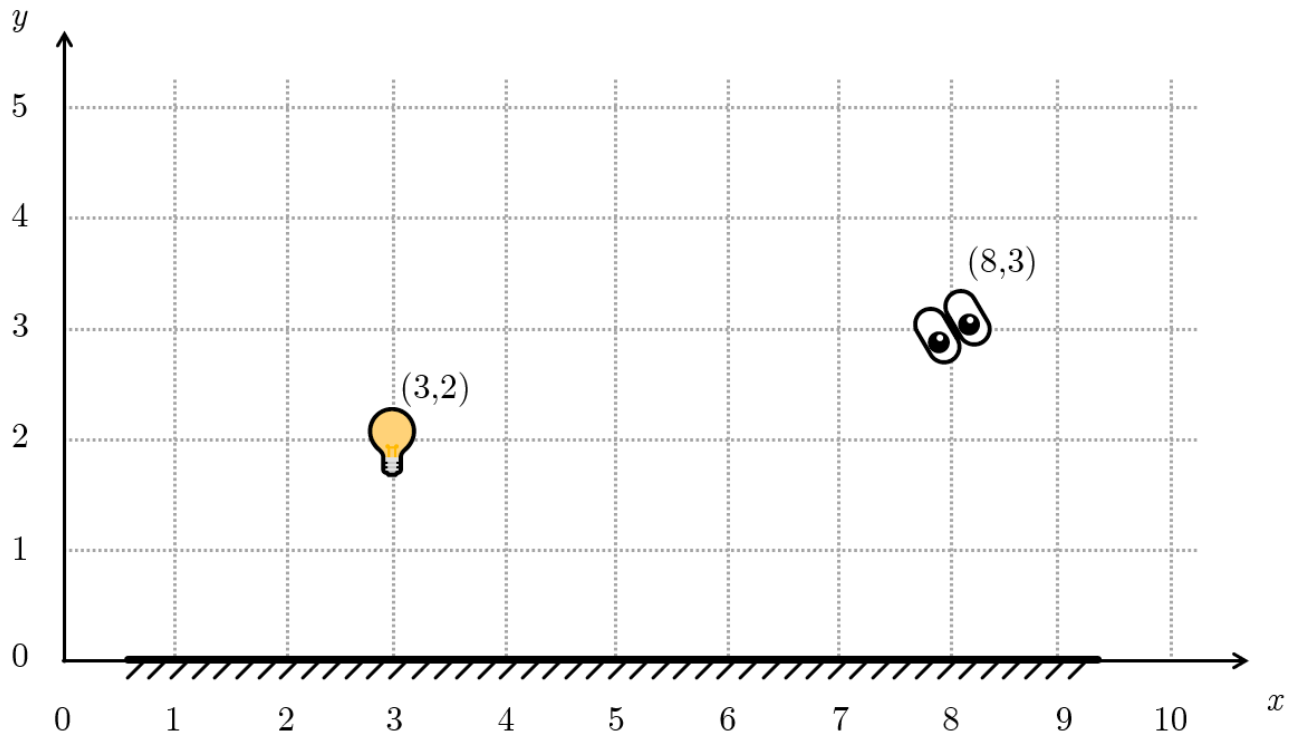
Quelles sont les valeurs d'illuminations à P_1 ($x=-3$), P_2 ($x=4$) et à $x=0$? Répondez avec deux décimales.

Selon le modèle de Gouraud à P_1 : ✓ , à P_2 : ✓ et à $x=0$: ✗

Selon le modèle de Phong à P_1 : ✓ , à P_2 : ✓ et à $x=0$: ✗

Description

Dans la scène 2D ci-dessous, il y a une surface en $y=0$, une source lumineuse ponctuelle à la position $(3,2)$ et un observateur à la position $(8,3)$. Le vecteur normal de la surface est dirigé vers le haut et le même sur toute la surface. On suppose que la surface en $y=0$ reflète toute la lumière spéculaire et aucune lumière ambiante ou diffuse. Il n'y a aucune atténuation d'intensité selon la distance.



Question 4.1

Partiellement correct

Note de 0,50 sur 1,00

Donnez à quelle position en x est le point de la surface qui apparaît le plus éclairé (du point de vue de l'observateur).

Selon le modèle spéculaire de Blinn (une décimale est suffisante) : ❌

Selon le modèle spéculaire de Phong : ✅

Question 4.2

Incorrect

Note de 0,00 sur 0,75

Comment se déplacerait le point le plus éclairé de la surface par rapport à votre réponse à la première sous-question si on repositionne l'observateur à (8, 10)? On applique le modèle de Blinn.

- ☒ a. Aucun changement. ✖
- ☐ b. Déplace vers la gauche (vers les x négatifs)
- ☐ c. Déplace vers la droite (vers les x positifs).

Votre réponse est incorrecte.

La réponse correcte est :

Déplace vers la gauche (vers les x négatifs)

Question 5

Partiellement correct

Note de 0,38 sur 0,75

Parmi les déclarations suivantes sur les **nuanceurs de géométrie**, cochez celles qui sont **vraies** :

- ☐ a. Le nuanceur est appelé une fois par sommet.
- ☒ b. Le nuanceur ne peut pas produire des lignes ou triangles disjoints, car les types de sortie sont limités à *points*, *line_strip* et *triangle_strip*. ✖
- ☐ c. Le paramètre *max_vertices* spécifie le nombre exact de sommets émis par le nuanceur.
- ☒ d. Le nuanceur est appelé une fois par primitive. ✔
- ☒ e. Le nuanceur peut émettre un nombre de sommets différent de celui de la primitive d'entrée. ✔

Votre réponse est partiellement correcte.

Vous avez sélectionné trop d'options.

Les réponses correctes sont :

Le nuanceur est appelé une fois par primitive.,

Le nuanceur peut émettre un nombre de sommets différent de celui de la primitive d'entrée.

Question 6

Partiellement correct

Note de 0,33 sur 0,50

Cochez le ou les cas dans lesquels l'utilisation d'un nuanceur de tessellation est particulièrement appropriée.

- ☒ a. Éclairage mieux défini selon le modèle de Lambert. ✓
- ☒ b. Raffiner une primitive dans un plan sans illumination. ✗
- ☐ c. Raffiner l'application d'une texture.
- ☒ d. Générer dynamiquement la géométrie d'un terrain à partir d'une texture topographique (*heightmap*) ✓
- ☒ e. Adapter le niveau de détail géométrique selon la distance à la caméra. ✓

Votre réponse est partiellement correcte.

Vous avez sélectionné trop d'options.

Les réponses correctes sont :

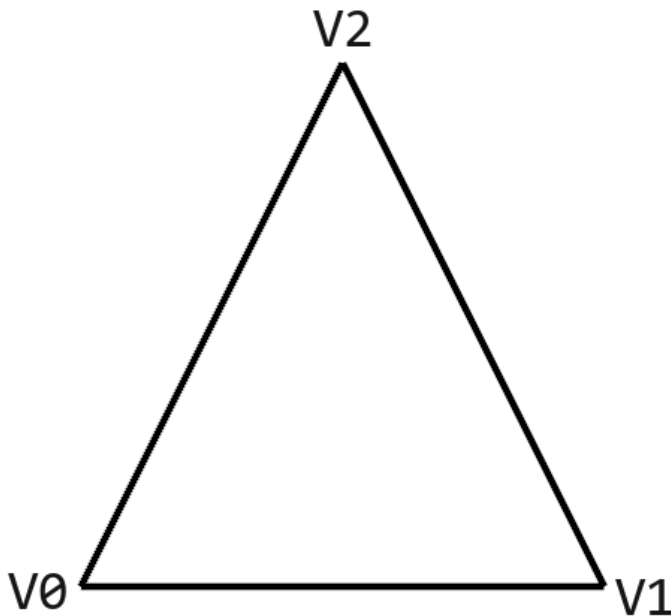
Générer dynamiquement la géométrie d'un terrain à partir d'une texture topographique (*heightmap*),

Éclairage mieux défini selon le modèle de Lambert.,

Adapter le niveau de détail géométrique selon la distance à la caméra.

Description

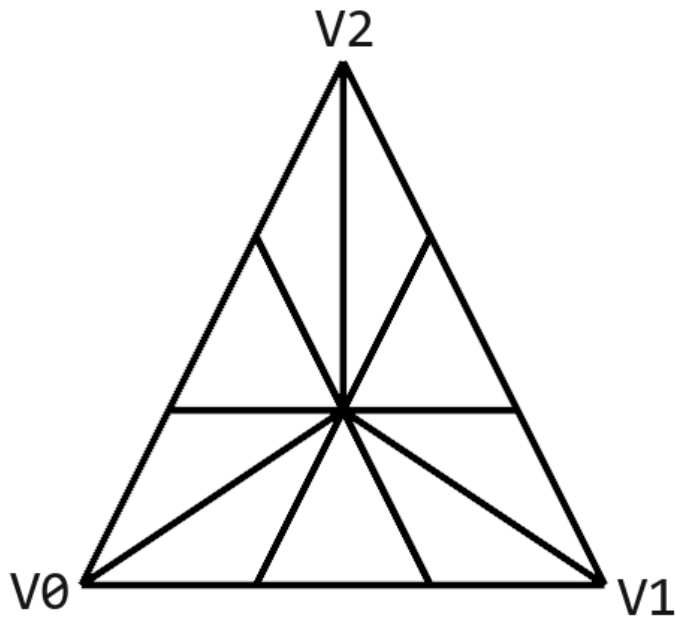
Soit le triangle défini par les sommets V0, V1 et V2 dans les images suivantes. Quels sont les niveaux de tessellation (intérieur et extérieur) appliqués à cette primitive pour obtenir chacune des formes dans les images suivantes? On suppose aucun repositionnement des sommets.



Question 7.1

Partiellement correct

Note de 0,25 sur 0,50

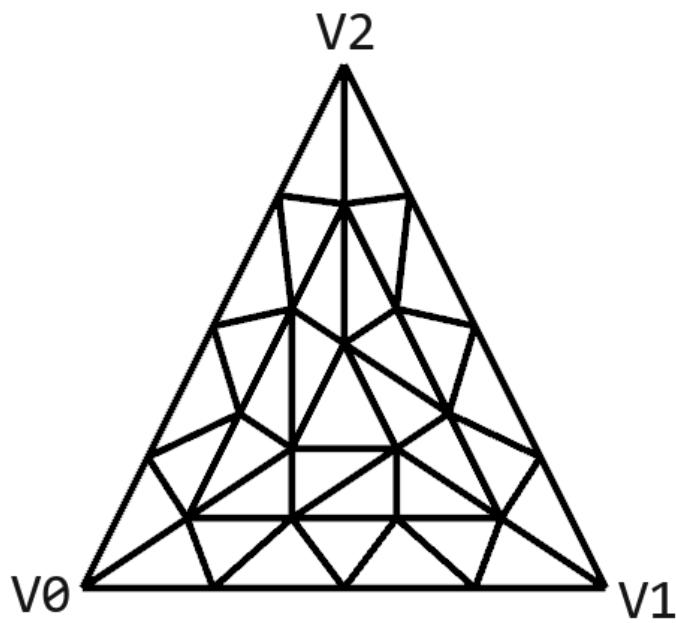


Niveaux intérieurs : ✓ , Niveaux extérieurs : ✗

Question 7.2

Incorrect

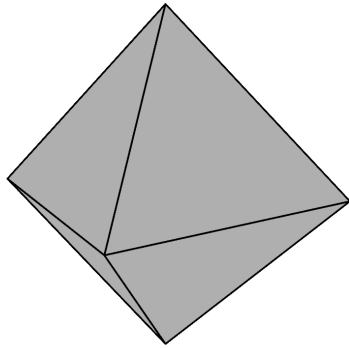
Note de 0,00 sur 0,75



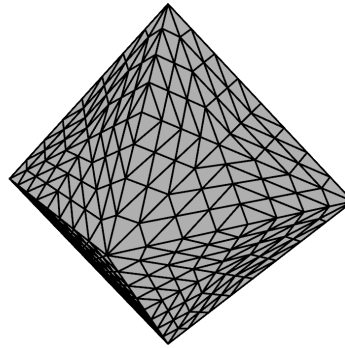
Niveaux intérieurs : ✖ , Niveaux extérieurs : ✖

Description

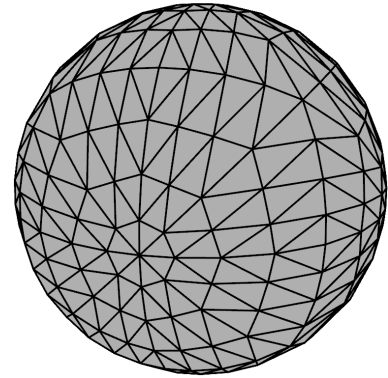
Les nuanceurs de tessellation permettent de modifier le rendu d'un simple octaèdre (image i) en appliquant un certain niveau de tessellation (image ii). Avec ces nuanceurs, si on repositionne tous les sommets introduits par la tessellation à une distance égale du centre de l'octaèdre, on peut alors afficher une sphère (image iii).



i) Octaèdre de base



ii) ... avec tessellation



iii) ... avec tessellation et repositionnement

On veut dessiner une sphère de rayon 1 centrée à l'origine (comme à l'image iii) et on suppose que l'octaèdre de base est régulier et aussi centré à l'origine. Ci-dessous, nous avons un bout pertinent du nuanceur d'évaluation de la tessellation qui tente de dessiner la sphère :

```
layout(triangles) in;

...

void main() {
    // 1. Obtenir les positions des sommets originaux de la patch triangulaire.
    vec3 v0=..., v1=..., v2=...;

    // 2. Calculer la position du sommet courant en coordonnées d'objet.
    vec3 posObjet = moyennePondérée(v0, v1, v2);

    // 3. Repositionner sur la surface d'une sphère de rayon 1.
    gl_Position = ...

    // 4. Appliquer les transformations usuelles.
    gl_Position = proj * visu * modèle * gl_Position;
}
```

Question 8.1

Correct

Note de 0,75 sur 0,75

Le repositionnement (troisième énoncé dans le *main*) est à compléter de façon à obtenir une sphère comme à l'image iii. On suppose que l'interpolation pondérée de la position du sommet est correcte, que les nuanceurs sont bien chargés et qu'il n'y a pas d'erreurs de syntaxe.

Parmi les choix suivants, cochez celui qui applique correctement le repositionnement de sommet pour produire l'effet escompté.

- ☒ a. `gl_Position = vec4(normalize(posObjet), 1);` ✓
- ☐ b. `gl_Position = normalize(vec4(posObjet, 1));`
- ☐ c. `gl_Position = vec4(posObjet, 1);`
- ☐ d. `gl_Position = vec4(normalize(posObjet), length(posObjet));`

Votre réponse est correcte.

La réponse correcte est :

```
gl_Position = vec4(normalize(posObjet), 1);
```

Question 8.2

Correct

Note de 1,00 sur 1,00

Dans l'exemple précédent, on dessinait une sphère de rayon 1, et ce peu importe la taille de l'octaèdre de base. On voudrait que le rayon de la sphère dépende de la taille de l'octaèdre tel qu'il est passé au pipeline sans avoir recours à une variable uniforme. On suppose toujours que l'octaèdre est centré à l'origine.

Parmi les bouts de codes suivants, lequel permet d'appliquer le rayon de l'octaèdre si on l'insère entre les étapes 3 et 4 (après le repositionnement).

- ☒ a.

```
float rayonMoyen = moyennePondérée(length(v0), length(v1), length(v2));  
gl_Position.xyz = rayonMoyen * gl_Position.xyz;
```

 ✓
- ☐ b.

```
float rayonMoyen = length(moyennePondérée(v0, v1, v2));  
gl_Position.xyz = rayonMoyen * gl_Position.xyz;
```
- ☐ c.

```
float rayonMoyen = moyennePondérée(length(v0), length(v1), length(v2));  
gl_Position.w = rayonMoyen;
```

Votre réponse est correcte.

La réponse correcte est :

```
float rayonMoyen = moyennePondérée(length(v0), length(v1), length(v2));  
gl_Position.xyz = rayonMoyen * gl_Position.xyz;
```

Question 9

Correct

Note de 0,50 sur 0,50

Lorsqu'on exécute un programme GLSL en mode de rétroaction avec un *transform feedback object*, que fait-on généralement dans le nuanceur de fragments?

- ☒ a. On l'ignore en désactivant le tramage. ✓
- ☐ b. On utilise le tampon de couleur pour rapatrier les résultats avec glReadPixels
- ☐ c. On sort les résultats des nuanceurs dans les valeurs de sortie des fragments.

Votre réponse est correcte.

La réponse correcte est :

On l'ignore en désactivant le tramage.

Question 10

Correct

Note de 0,75 sur 0,75

Soit la capture d'écran suivante d'un jeu vidéo. Dans cette capture, on a encadré en rouge la mini-carte fixe en haut à gauche de l'écran et l'affichage du radar qui est plaqué sur la géométrie du moniteur. Ces deux éléments sont dynamiques et dépendent des objets dans la scène et sont donc mis à jour à chaque trame. Ils ont donc chacun leur propre programme nuanceur qui les dessine.



(Jeu : Raft, développé par Redbeet Interactive et publié par Axolot Games.)

Quelle(s) méthode(s) pourrait-on raisonnablement utiliser pour afficher l'écran du radar sur la surface de son moniteur?

- ☒ Texture de rendu appliquée sur un objet de la scène ✓
- ☐ Appel à glViewport pour positionner et dimensionner sur le moniteur du radar

Note de 1,00 sur 1,00

La réponse correcte est :

- Texture de rendu appliquée sur un objet de la scène

Quelle(s) méthode(s) pourrait-on raisonnablement utiliser pour afficher la mini-carte fixe dans le HUD?

- ☒ Texture de rendu appliqué sur un quad dans le HUD ✓
- ☒ Appel à glViewport pour écrire sur l'écran par-dessus la scène ✓

Note de 1,00 sur 1,00

La réponse correcte est :

- Texture de rendu appliqué sur un quad dans le HUD

- Appel à glViewport pour écrire sur l'écran par-dessus la scène

Question 11

Correct

Note de 0,50 sur 0,50

Soit deux B-Splines dont les enveloppes convexes des points de contrôles se superposent. Que peut-on conclure sur l'intersection de ces courbes?

- ☐ a. Il est **certain** que les courbes se croisent
- ☒ b. Il est **possible** que les courbes se croisent ✓
- ☐ c. Il est **impossible** que les courbes se croisent

Votre réponse est correcte.

La réponse correcte est :

Il est **possible** que les courbes se croisent

Question 12

Correct

Note de 1,50 sur 1,50

Les courbes paramétriques sont définies par la relation $C(t) = \sum_{i=1}^n N_{i,k}(t)G_i$ où $N(t)$ est la fonction de base et G_i est le vecteur d'informations géométriques. Associez le type de courbes à la fonction de base.

A) $N_{i,k=n}(t) = \binom{n}{i} t^i (1-t)^{n-1}$

Bézier



B) $N_{i,k=4}(t) = H_i(t)$, où

$$H_1(t) = 2t^3 - 3t^2 + 1$$

$$H_2(t) = -2t^3 + 3t^2$$

$$H_3(t) = t^3 - 2t^2 + t$$

$$H_4(t) = t^3 - t^2$$

Spline cubique



C) $N_{i,k}(t) = \frac{h_i N_{i,k}^*(t)}{\sum_{j=1}^n h_j N_{j,k}^*(t)}$

où $N_{i,k}^*(t) = \frac{t-u_i}{u_{i+k-1}-u_i} N_{i,k-1}^*(t) + \frac{u_{i+k}-t}{u_{i+k}-u_{i+1}} N_{i+1,k-1}^*(t)$

et $N_{i,1}^*(t) = \begin{cases} 1 & \text{si } u_i \leq t \leq u_{i+1} \\ 0 & \text{sinon} \end{cases}$

NURBS



D) $N_{i,k}(t) = \frac{t-u_i}{u_{i+k-1}-u_i} N_{i,k-1}(t) + \frac{u_{i+k}-t}{u_{i+k}-u_{i+1}} N_{i+1,k-1}(t)$

où $N_{i,1}(t) = \begin{cases} 1 & \text{si } u_i \leq t \leq u_{i+1} \\ 0 & \text{sinon} \end{cases}$

B-Spline

**Question 13**

Correct

Note de 0,50 sur 0,50

Quel type de courbe parmi les suivants est le plus général? C'est-à-dire celui qui permet de reproduire les autres en paramétrisant d'une façon spécifique.

- ☐ a. Bézier
- ☒ b. NURBS
- ☐ c. B-Spline

Votre réponse est correcte.

La réponse correcte est :

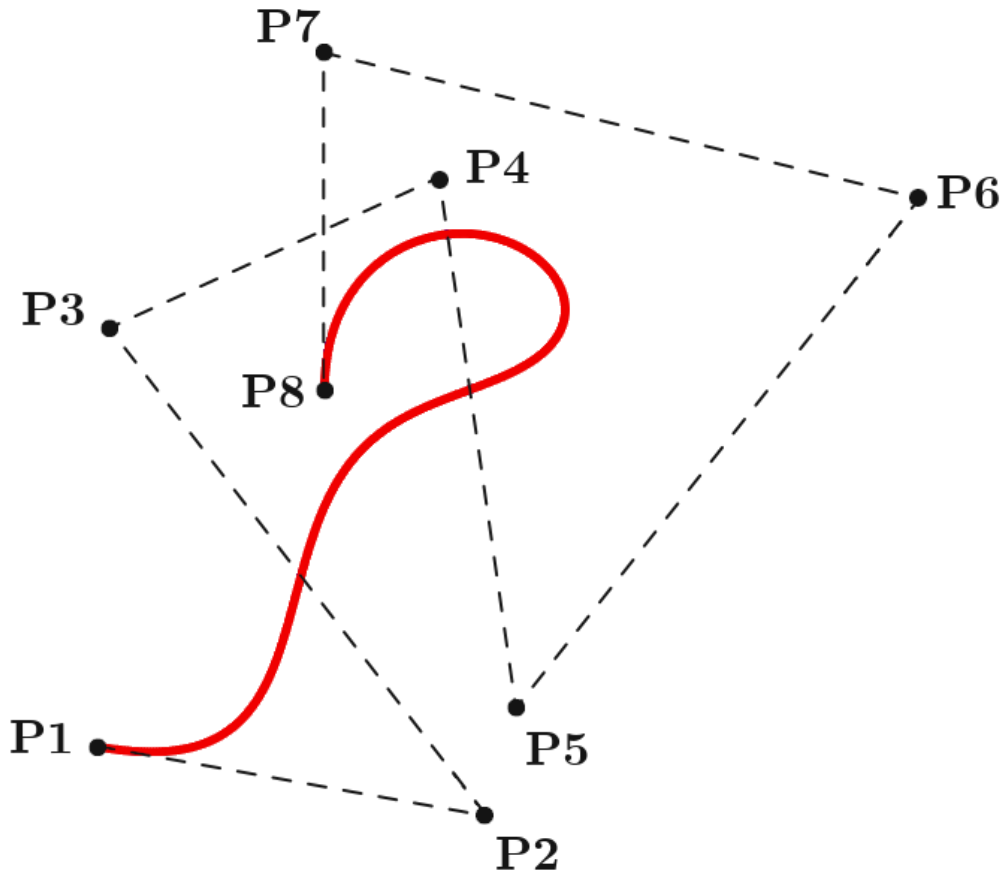
NURBS

Question 14

Correct

Note de 0,75 sur 0,75

Soit l'image suivante dans laquelle on a les points de contrôle P1 à P8 et une courbe paramétrée selon ceux-ci. Quel type de courbe a été utilisé pour dessiner la courbe rouge?



- ☐ a. Spline quadratique
- ☒ b. Bézier ✓
- ☐ c. Spline cubique

Votre réponse est correcte.

La réponse correcte est :
Bézier

Question 15

Correct

Note de 0,75 sur 0,75

Soit les quatres points de contrôle suivants qui sont utilisés dans cet ordre :

$$P_1 = (0,0)$$

$$P_2 = (0,1)$$

$$P_3 = (1,1)$$

$$P_4 = (1,-1)$$

Parmi les courbes suivantes qui utilisent ces points de contrôles, choisissez celle qui passe le plus près du point (1,1).

- ☒ a. Une NURBS avec $n=4$, $k=4$, le vecteur nodal $[0,0,0,0,1,1,1,1]$ et le vecteur de coordonnées homogènes $[1,2,5,1]$. ✓
- ☐ b. Une courbe de Bézier cubique.
- ☐ c. Une B-Spline avec $n=4$, $k=4$ et le vecteur nodal $[0,0,0,0,1,1,1,1]$.

Votre réponse est correcte.

La réponse correcte est :

Une NURBS avec $n=4$, $k=4$, le vecteur nodal $[0,0,0,0,1,1,1,1]$ et le vecteur de coordonnées homogènes $[1,2,5,1]$.

Question 16

Correct

Note de 0,50 sur 0,50

Lequel des types de courbe est le mieux adapté pour représenter un arc elliptique?

- ☐ a. Spline cubique
- ☒ b. NURBS ✓
- ☐ c. B-Spline
- ☐ d. Bézier

Votre réponse est correcte.

La réponse correcte est :

NURBS

Question 17

Correct

Note de 0,75 sur 0,75

L'algorithme de Bresenham sert à créer efficacement les fragments d'une droite dans une grille de pixels. Le pseudo-code qu'on a dans les notes de cours (rappel ci-dessous) permet de traiter les droites dont le premier point (x1,y1) est à gauche du deuxième point (x2,y2) et dont la pente est entre 0 et 1. Quelles sont les symétries qu'on peut utiliser pour traiter les autres cas?

Premier point à droite du deuxième (x1 > x2) : Inverser le point de départ et le point d'arrivée ✓

Pente plus grande que 1 : Balayer par rapport à y plutôt qu'à x ✓

```
FONCTION Bresenham(x1, y1, x2, y2)
  dx = x2 - x1
  dy = y2 - y1
  d = 2*dy - dx
  CreerFragment(x1, y1)
  TANT QUE x1 < x2
    SI d <= 0 ALORS // EST
      d = d + 2*dy
    SINON // NORD-EST
      d = d + 2*(dy-dx)
      y1 = y1 + 1
      x1 = x1 + 1
    CreerFragment(x1, y1)
```

Question 18

Correct

Note de 0,50 sur 0,50

Dans le langage PostScript, l'unité de mesure de base des coordonnées est le *point*. À quoi correspond un point en PostScript?

- ☐ a. 1 point = 1 pixel
- ☒ b. 1 point = 1/72 pouce ✓
- ☐ c. 1 point = 72 pixels
- ☐ d. 1 point = 72 pouces

Votre réponse est correcte.

La réponse correcte est :

1 point = 1/72 pouce

Question 19

Correct

Note de 0,50 sur 0,50

Soit le code suivant qui charge un programme nuanceur en WebGL. Ce code est clairement en Javascript, mais dans quel langage est écrit le code source des nuanceurs en WebGL?

```
var canvas = document.getElementById("canvas");
var gl = canvas.getContext("webgl");
// ...
var srcVert = ...; // Code source du nuanceur de sommets.
var srcFrag = ...; // Code source du nuanceur de fragments.
var prog = gl.createProgram();
// Charger le nuanceur de sommets.
var vertShader = gl.createShader(gl.VERTEX_SHADER);
gl.shaderSource(vertShader, srcVert);
gl.compileShader(vertShader);
gl.attachShader(prog, vertShader);
// Charger le nuanceur de fragments.
var fragShader = gl.createShader(gl.FRAGMENT_SHADER);
gl.shaderSource(fragShader, srcFrag);
gl.compileShader(fragShader);
gl.attachShader(prog, fragShader);
// Lier et utiliser le programme.
gl.linkProgram(prog);
gl.useProgram(prog);
```

- ☐ a. Web Assembly
- ☐ b. C++
- ☒ c. GLSL ✓
- ☐ d. Javascript

Votre réponse est correcte.

La réponse correcte est :

GLSL