Question 1 Notions de base [11 points]

a) [2 points] Quelle est la différence principale entre une projection orthographique et une projection perspective ?

La projection orthographique conserve la forme des objets c'est-à-dire qu'elle garde les même rapports de proportionnalités. Tandis que la projection perspective peut déformer l'objet selon son angle d projection

2

b) [2 points] En infographie, qu'est-ce qu'un *lutin* et quel est l'intérêt d'en utiliser dans un logiciel d'infographie ?

Un lutin en Infographie est representation d'un objet 3D sur un quadrilatère. L'intérêt d'utiliser un lutin réside en ce qu'il évite de dessiner les objets 3D qui demande des calculs très complexes.

2

c) [3 points] Dans le pipeline graphique programmable, nous avons utilisé le produit des matrices de modélisation, de visualisation et de projection par le sommet courant afin d'obtenir une position à l'écran. Donnez *deux* exemples distincts qui illustrent l'utilité d'avoir accès, dans les nuanceurs, aux trois matrices séparément plutôt qu'à une seule matrice qui contiendrait le produit des trois. Justifiez vos réponses.

1) Pour la tesselation par exemple ~~pour~~ on peut on peut faire la projection et la animation dans le nuanceur de contrôle de tesselation plutôt que dans le nuanceur de sommet

1.5

2)

d) [4 points] Nous avons vu que les nuanceurs de tessellation permettent de subdiviser une primitive en y générant des triangles intermédiaires. Les sous-triangles ainsi générés ajoutent toutefois au temps de traitement nécessaire pour l'affichage de la scène.

i) Donnez *deux* exemples, autres que celui du TP3, où il est utile et pertinent de raffiner ainsi l'affichage d'une primitive. Justifiez vos réponses.

1) On peut raffiner l'affichage d'un triangle. Plutôt que d'avoir un gros triangle, il peut être ~~utile~~ utile de ~~le~~ subdiviser en plusieurs sous-triangles.

2) On peut raffiner l'affichage d'une sphère ce qui évite de remarquer certaines discontinuités à la surface.

4

ii) À l'inverse, donnez *deux* exemples où il est inutile et contreproductif de raffiner ainsi l'affichage d'une primitive. Justifiez vos réponses.

1) il est inutile de vouloir raffiner un point, en effet un point est constitué d'un seul sommet donc n'a pas besoin d'un sommet supplémentaire pour être raffiner

2) Une ligne n'a pas besoin de raffiner son affichage car elle est entièrement définir par 2 points

Question 2 Illumination [14 points]

a) [8 points] Dans le modèle de réflexion locale de la lumière communément utilisé en infographie, trois sortes de réflexion sont définies : *ambiante*, *diffuse* et *spéculaire*. Pour chaque élément ci-dessous, identifiez toutes les réflexions (parmi ces trois) sur laquelle ou lesquelles cet élément a une influence ; écrivez « *aucune* » si l'élément n'a aucune influence.

- Le vecteur normal à la surface. diffuse, spéculaire
- Le coefficient de brillance. spéculaire
- La position de la source lumineuse. diffuse, spéculaire
- La position de l'observateur. spéculaire
- La position des autres objets de la scène. aucune
- Le nombre total d'objets dans la scène. aucune
- Les propriétés de matériau de l'objet. ambiante, diffuse, spéculaire
- Un autre objet situé entre l'objet éclairé et la source lumineuse. diffuse, spéculaire X

7

b) [1 point] Lorsqu'une source lumineuse d'OpenGL éclaire une surface, dans quelle direction est réfléchie la lumière diffuse ?

La lumière diffuse est reflechie selon la normale à la surface X

0

c) [1 point] Lorsqu'une source lumineuse d'OpenGL éclaire une surface, dans quelle direction est réfléchie la lumière ambiante ?

La lumière ambiante est reflechie dans toutes les directions

1

d) [4 points] Dans la dernière partie du TP2 avec les poissons-théières, les vecteurs vers la source de lumière et vers l'observateur étaient constants :

```
lumiDir = vec3( 0, 0, 1 );   obsVec = vec3( 0, 0, 1 );
```

L'énoncé mentionnait rapidement que « *ces simplifications pour la position de la lumière et pour la position de l'observateur sont souvent utilisées pour que les calculs d'illumination soient plus simples* ».

Expliquez pourquoi les calculs d'illumination sont alors plus simples.

4

On a ~~$I = K_a I_a + K_d I_d (\vec{L}\cdot\vec{N})$~~

$I_\Sigma = I_a + I_d + I_s$ avec $I_a = K_a I_a$

$I_d = K_d I_d (\vec{L}\cdot\vec{N})$

et $I_s = K_s I_s (\vec{R}\cdot\vec{O})^n$ selon Phong

ou $I_s = K_s I_s (\vec{B}\cdot\vec{N})$ selon Blinn

Pour $\vec{L} = vec(0,0,1)$ alors la lumière diffuse se calcule seulement avec la composante z normalisé de N alors $\begin{cases} I_d = K_d I_d \text{ si } \vec{N} = \vec{z} \\ I_d = 0 \text{ sinon} \end{cases}$

on a $\vec{R} = 2(\vec{L}\cdot\vec{N})\vec{N} - \vec{L} = 2\vec{N} - \vec{L}$

$\vec{B} = (\vec{L}+\vec{O}) / \|\vec{L}+\vec{O}\| = \frac{(0,0,2)}{\sqrt{2}}$

$\vec{R}\cdot\vec{O} = (2\vec{N} - \vec{L})\cdot\vec{O} = 2N_z - 1$ et $\vec{B}\cdot\vec{N} = \frac{2}{\sqrt{2}} N_z$

$I_s = \begin{cases} I_s K_s & \text{si } \vec{N} = \vec{z} \text{ / Phong} \\ 0 & \text{sinon} \end{cases}$

$I_s = \begin{cases} K_s K_s \left(\frac{2}{\sqrt{2}}\right)^n & \text{si } N = \vec{z} \text{ Blinn / On ~~fait~~ calcul seulement illumination si } \vec{N} = \vec{z} \\ 0 & \text{sinon} \end{cases}$

Question 3 Nuanceurs en GLSL [12 points]

a) [5 points] Dans le cadre d'un projet en infographie, on vous donne un programme avec les nuanceurs listés ci-dessous (cette page et la suivante). Vous constatez que le code GLSL est assez mal documenté et que les noms de variables ne sont pas toujours explicites. :(

Écrivez les dix (10) commentaires manquants dans les `main()` des nuanceurs de sommets et de fragments. Afin de montrer que vous comprenez ce que fait l'énoncé, débutez avec un verbe à l'infinitif et décrivez bien le but de l'énoncé dans votre commentaire. (Par exemple, en référence au dernier TP avec les lutins, préférez « *// appliquer la gravité à cette particule* » plutôt que simplement « *// calculer la vitesse* ».)

i) Le nuanceur de sommets `"nuanceurSommets.glsl"` :

```
#version 410
layout (std140) uniform LightSourceParameters
{
  vec4 ambient, diffuse, specular, position;
} LightSource;

uniform mat4 matrModel, matrVisu, matrProj;
uniform mat3 matrNormale;

layout(location=0) in vec4 Vertex;
layout(location=2) in vec3 Normal;
layout(location=3) in vec4 Color;

out Attribs { vec3 l, n, o; } AttribsOut;

void main(void)
{

  // ? La position des sommets dans le repère de la camera
  gl_Position = matrProj * matrVisu * matrModel * Vertex;


  // ? Les coordonnées (x,y,z) d'un sommet   lequel ? et pourquoi ?
  vec3 p = ( matrVisu * matrModel * Vertex ).xyz;


  // ? Assigner la normale pour le nuanceur de fragment
  AttribsOut.n = matrNormale * Normal;


  // ? Assigner la direction de la lumière
  AttribsOut.l = ( matrVisu * LightSource.position ).xyz - p;


  // ? Assigner la position de l'observateur
  AttribsOut.o = normalize(-p);
}
```

ii) Le nuanceur de fragments "nuanceurFragments.glsl" :

```
#version 410
layout (std140) uniform LightSourceParameters
{
  vec4 ambient, diffuse, specular, position;
} LightSource;

layout (std140) uniform MaterialParameters
{
  vec4 emission, ambient, diffuse, specular;
  float shininess;
} FrontMaterial;

in Attribs { vec3 l, n, o; } AttribsIn;

out vec4 FragColor;

void main(void)
{
    vec3 L = normalize( AttribsIn.l );
    vec3 N = normalize( AttribsIn.n );
    vec3 O = normalize( AttribsIn.o );
    Vec3 B = normalize (L+O)

    // ? Calcul l'intensité de la lumière ambiante
    vec4 c = FrontMaterial.ambient * LightSource.ambient;


    // ? Le facteur d'atténuation
    float d = length( AttribsIn.l );
    float a = min( 1.0, 1.0 / ( 0.01 + 0.05 * d + 0.1 * d * d ) );


    // ? Calcul du produit scalaire L.N (si L.N < 0 alors f=0)
    float f = max( 0.0, dot( N, L ) );
    float s = max(0.0, dot(B,N)) , // question D)

    // ? On Calcul la lumière diffuse qu'on ajoute à la lumière ambiante
    c += a * FrontMaterial.diffuse * LightSource.diffuse * f;
    c+ = a* FrontMaterial.spécular*LightSource.spécular*pow(s,FrontMaterial.shininess)

    // ? Assigner la couleur au fragment
    FragColor = clamp( c, 0.0, 1.0 );
}
```

4.5

b) **[1 point]** Comment nomme-t-on le *modèle d'illumination* implémenté dans les nuanceurs de la sous-question précédente ?

1 il s'agit du modèle de Phong

c) **[3 points]** Quel rendu visuel produit habituellement le *modèle d'illumination* implémenté dans ces nuanceurs, le rendu A ou le rendu B ? Justifiez succinctement votre raisonnement.

rendu A : 

rendu B : 

3 il s'agit du rendu A car le modèle de phong interpole la normal en chaque point de une face plutôt qu'une normale par face (cas rendu B)

d) **[3 points]** Dans les nuanceurs fournis, aucune réflexion spéculaire n'est calculée. Écrivez les énoncés manquants dans ces nuanceurs afin d'ajouter le calcul de la réflexion spéculaire selon Blinn, ce qui donnera les rendus C ou D ci-dessous. Indiquez clairement entre quelles lignes s'insèrent vos ajouts.

rendu C : 

rendu D : 

2 Voir le nuanceur question précédente

d'où vient B ?

Question 4 Utilisation des textures [7 points]

Un petit logiciel graphique charge les deux images "scene.bmp" et "neige.bmp" (voir ci-dessous) dans deux unités de texture distinctes et trace deux triangles formant un carré. Les deux textures sont utilisées sur ce carré pour produire l'image *Scène complètement enneigée* (à droite ci-dessous).

scene.bmp

neige.bmp

Scène complètement enneigée

a) [3 points] Le programme principal `main.cpp` ainsi que le nuanceur de sommets ont fait tout ce qui est nécessaire pour le bon fonctionnement de l'application. Complétez le nuanceur de fragments en GLSL afin de produire le résultat désiré. (Notez bien que la scène n'est pas plus sombre lorsqu'enneigée.)

```
#version 410
uniform sampler2D laTextureScene, laTextureNeige;
in Attribs { vec2 texCoord; } AttribsIn;
out vec4 FragColor;
void main( void )
{
  // vec4 texture( sampler2D sampler, vec2 coo );
// On a la texture de la scène entièrement
 FragColor = texture(laTextureScène, AttribIn.texCoord)
 vec4 c = texture(laTextureNeige, AttribIn.texCoord)

 FragColor += c
}
```

3

b) **[2 points]** On souhaite donner l'illusion que les flocons de neige tombent, c'est-à-dire que la position des flocons varie en fonction du temps, comme illustré dans les images *Carrés texturés de façon variable selon le temps* ci-dessous.

Carrés texturés de façon variable selon le temps (à t=0.0, t=0.25, t=0.5, t=0.85, t=1.0 seconde).

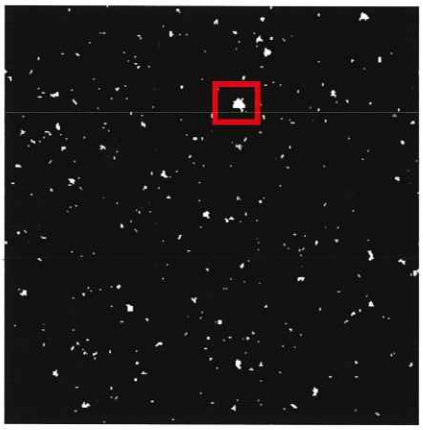
L'animation fait en sorte que les flocons de neige descendent régulièrement, disparaissent en bas de l'image pour réapparaître en haut, et continuent à descendre régulièrement. Dans ces images, observez le déplacement du flocon entouré d'un rectangle rouge : il descend régulièrement puis revient en haut pour continuer sa descente. Un flocon de neige repasse ainsi à sa position initiale chaque seconde.
En vue de produire l'animation avec les nuanceurs, le programme principal transmet aux nuanceurs la variable `uniform float temps` (en secondes), initialisée à `0` et incrémentée de façon régulière par le programme principal.

Écrivez les modifications à faire au nuanceur de fragments de la sous-question précédente (ou réécrivez-le) afin de produire le résultat désiré.

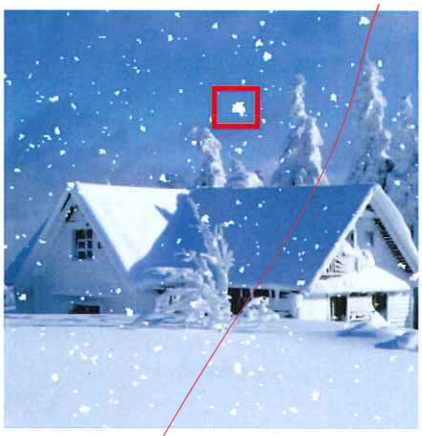
```
#version 410
uniform float temps;
uniform sampler2D laTextureScene, laTextureNeige;
in Attribs { vec2 texCoord; } AttribsIn;
out vec4 FragColor;
void main( void )
{
   Vec2 Coord = AttribIn.texCoord;
   FragColor= texture(Coord, laTextureScene);
   Coord.y- = temps * Coord.y;
     if(Coord.y>0)
     { FragColor+ = texture(Coord, laTextureNeige);
     }

}
```

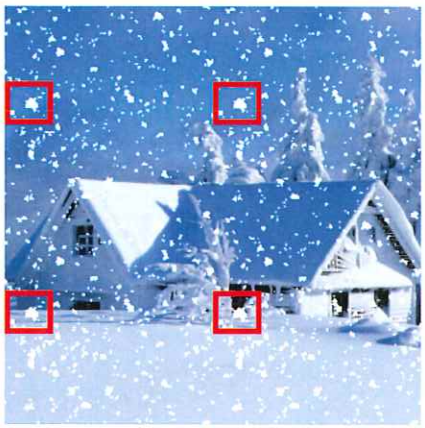
c) [2 points] En utilisant toujours les mêmes textures et le même programme principal et nuanceur de sommets, on souhaite afficher des flocons de neige qui tombent (comme précédemment), mais avec une neige qui est deux fois plus dense, comme illustré dans l'image *Scène et neige plus dense* ci-dessous. (Notez que les flocons sont plus nombreux, mais ils n'ont pas changé de taille.)



neige.bmp



Scène et neige (version originale)



Scène et neige plus dense

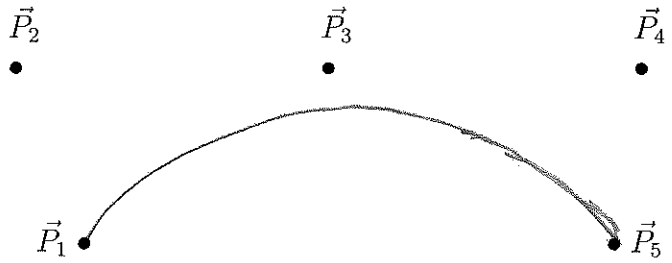
Écrivez les modifications à faire au nuanceur de fragments de la sous-question précédente (ou réécrivez-le) afin de produire le résultat désiré. (Indice : le même flocon encadré de rouge est présent à quatre endroits.)

```
#version 410
uniform float temps;
uniform sampler2D laTextureScene, laTextureNeige;
in Attribs { vec2 texCoord; } AttribsIn;
out vec4 FragColor;
void main( void )
{
    vec2 Coord = AttribsIn.texCoord;
    FragColor = texture(Coord, laTextureScène);
    Coord = 4*Coord
    FragColor += texture(Coord, laTextureNeige);
}
```

Question 5 Courbes paramétriques polynomiales [6 points]

a) [3 points] Considérez cinq points de contrôle $\{\vec{P}_1, \vec{P}_2, \vec{P}_3, \vec{P}_4, \vec{P}_5\}$ utilisés pour construire une courbe de Bézier.

i) Dessinez la courbe de Bézier utilisant ces cinq points de contrôle.



2

ii) Par lequel ou lesquels des cinq points de contrôle la courbe de Bézier passe-t-elle certainement ? Pourquoi ?

La Courbe passe Certainement par les point P_1 et P_5 car le premier point et le dernier point appartient toujour à la Courbe de Bezier

iii) Quelle(s) contrainte(s) devrait-on imposer à ces points de contrôle pour que la courbe de Bézier passe par les cinq points ?

X

b) **[2 points]** Identifiez deux différences fondamentales entre les courbes de Bézier et les splines cubiques.

1) Les splines cubiques passent par des points expérimentaux tandis que les courbes de Béziers ne passe pas forcément par tous ses point de contrôle

2

2) Les courbes de Béziers sont plus générales que les courbe splines cubique parce que les courbes de Beziers peuvent generer des courbes different en fonction des contrainte imposé aux point de contrôle

c) **[1 point]** Les fonctions de base d'une B-Spline satisfont toujours la relation $\sum_{i=1}^{n} N_{i,k}(t) = 1$ pour toute valeur de k ou de t. Expliquez en quoi cette propriété est utile.

Cette propriété est utile dans la mesure où elle favorise l'utisation des polynômes de Berstein

0

X

Bon hiver !

Cet examen comprend 5 questions sur 14 pages pour un total de 50 points.
Benoît Ozell

(Si vous utilisez cette page pour répondre à une question, inscrivez clairement le numéro de la question.)