NATIONAL UNIVERSITY OF SINGAPORE

CS3241 — COMPUTER GRAPHICS

(Semester 1 AY2020/2021)

Time Allowed: **2 Hours**

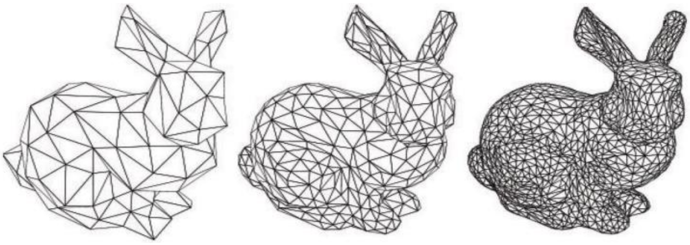
ANSWERS

INSTRUCTIONS

1. This assessment contains **25 questions** in **8 sections**.
2. The full score of this assessment is **80 marks**.
3. Answer **all questions**.
4. This is an **open-book** assessment.
5. **Follow the instructions of your invigilator or the module coordinator to submit your answers**.

Section A [7 marks]

In real-time rendering applications, such as in 3D games, it is very common to have multiple **level-of-detail** (LOD) polygon mesh representations of objects, such as the example shown below.



(1) [3 marks]

What is the **main reason** for having multiple **level-of-detail** (LOD) polygon mesh representations of objects?

It allows us to trade-off between rendering quality and rendering time of the object. For example, when the object looks very small on the screen, or when the frame time budget is limited, we can draw the lower LOD model to save time.

(2) [4 marks]

List **two issues/problems/drawbacks** that we may have when using multiple LOD representations of objects for rendering.

(1) Taking up more memory.

(2) Undesirable popping effect when switching between different LODs during animation.

Section B [10 marks]

(3) [3 marks]

When rendering **bump mapping**, should we perform **per-vertex** lighting computation or **per-fragment** lighting computation? Why?

We should use per-fragment lighting computation. To create the appearance of small surface details, every fragment potentially can have a different surface orientation from its neighboring fragments, and the effect of this difference is visible only if we perform lighting computation using the surface orientation at each fragment.

(4) [3 marks]

In Lab Assignment 3, what is the **main reason** we **subdivide** the table-top rectangle into many **smaller rectangles**?

To avoid missing the specular highlight.
To improve the quality/sharpness of the specular highlight.

(5) [4 marks]

The basic Phong Illumination Equation $I_{phong} = I_a k_a + I_p k_d (N \bullet L) + I_p k_s (R \bullet V)^n$ does not deal with the cases that the light source is "behind" the surface to be illuminated, and other "invalid" arrangements of the viewpoint and light source position with respect to the surface point. Which one of the followings is the *most* correct implementation of the Phong Illumination Equation to handle these "invalid" cases?

A. $I_{phong} = \max(\, 0,\ I_a k_a + I_p k_d (N \bullet L) + I_p k_s (R \bullet V)^n \,)$

B. $I_{phong} = I_a k_a + \max(\, 0,\ I_p k_d (N \bullet L) + I_p k_s (R \bullet V)^n \,)$

C. $I_{phong} = I_a k_a + I_p k_d \max(0, N \bullet L) + I_p k_s (R \bullet V)^n$

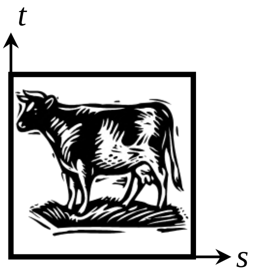
D. $I_{phong} = I_a k_a + I_p k_d (N \bullet L) + I_p k_s (\max(0, R \bullet V))^n$

E. $I_{phong} = I_a k_a + I_p k_d \max(0, N \bullet L) + I_p k_s (\max(0, R \bullet V))^n$

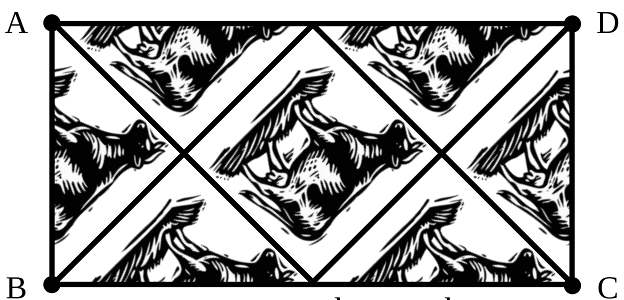
F. If $(N \bullet L < 0)$ then $\{ I_{phong} = I_a k_a \}$
else $\{ I_{phong} = I_a k_a + I_p k_d (N \bullet L) + I_p k_s (\max(0, R \bullet V))^n \}$ **(answer)**

Section C [6 marks]

Suppose the texture coordinate **wrapping mode** has been set to **`GL_REPEAT`** for both the *s* and *t* texture coordinates. Given a texture image and a rectangle ABCD as shown in the following diagram, we want to assign **2D texture coordinates** to the vertices so that the texture-mapped rectangle appears as shown below. It is given that the texture coordinates assigned to vertex **D** are (3, 6).



texture image



texture-mapped rectangle

(6) [2 marks]

What should be the **2D texture coordinates** assigned to vertex **A**?

(5, 4)

(7) [2 marks]

What should be the **2D texture coordinates** assigned to vertex **B**?

(6, 5)

(8) [2 marks]

What should be the **2D texture coordinates** assigned to vertex **C**?

(4, 7)

Section D [9 marks]

(9) [3 marks]

Given a texture image of size 2048 x 2048 texels, we want to create a **mipmap** from it. What would be the **number of levels** (including the original texture image) in the mipmap?

A. 1
B. 10
C. 11
D. 12 **(answer)**
E. 13
F. 1024
G. 2048

(10) [3 marks]

Suppose we have a **mipmap** constructed from an original texture image of size 2048 x 2048 texels, what would be the most **ideal mipmap level** to texture map a 3D square that appears in a 256 x 256 pixels region in the viewport?

A. 0
B. 2
C. 3 **(answer)**
D. 4
E. 7
F. 8
G. 9

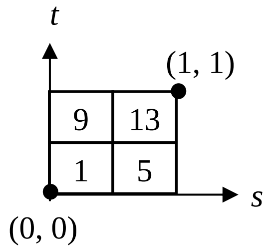
(11) [3 marks]

Consider a texture-mapped scene rendered using **mipmapping**. During rendering, suppose the mipmap level selection is always **biased by +2 levels**, how may the rendered result differ from the case when no bias is used?

A. There is no difference in the rendered result.
B. The resulting image will look darker in the textured regions.
C. The resulting image will look brighter in the textured regions.
D. The resulting image may have more aliasing artifacts.
E. The resulting image will look more blurred in the textured regions. **(answer)**
F. The resulting image will look sharper in the textured regions.

Section E [6 marks]

The following diagram depicts a **texture map** of size 2x2 texels. Each texel stores a **grayscale value**. The texture map is accessed using the texture coordinates **(3/8, 5/8)**.



(12) [2 marks]

Suppose the texture minification and magnification filtering have been set to **`GL_NEAREST`**, what will be the retrieved texture value? Round your answer to 2 decimal places.

9.00

(13) [4 marks]

Suppose the texture minification and magnification filtering have been set to **`GL_LINEAR`**, what will be the retrieved texture value? **Show your workings** and round your answer to 2 decimal places.

(9 - 1) / (0.75 - 0.25) = (a - 1) / ((5/8) - 0.25) ==> a = 7.00
(13 - 5) / (0.75 - 0.25) = (b - 5) / ((5/8) - 0.25) ==> b = 11.00
(b - a) / (0.75 - 0.25) = (c - a) / ((3/8) - 0.25) ==> c = 8.00

Answer: 8.00

Section F [13 marks]

Suppose there is an **opaque sphere** in an **enclosed** environment. All the other surfaces of the environment are **opaque** and have materials that have only **diffuse** component (i.e. $k_d > 0$, $k_r = 0$ and $k_{rg} = 0$). However, the sphere's material has both **diffuse and specular** components (i.e. k_d, k_r, and k_{rg} are all greater than 0). There are **two point light sources** in the scene.

We want to render a **300x200 pixels image** of the scene using **Whitted Ray Tracing**, with **two levels of recursion**.

Assume the camera is within the environment but outside the sphere, and the **sphere occupies 10,000 pixels** (i.e. number of primary rays that hit the sphere directly) in the rendered image.

In addition, at any surface point where lighting computation is to be performed, a **shadow ray** is always shot towards each light source even when $N \bullet L \leq 0$ (where N is the normal vector at the surface point and L is the vector towards the light source).

(14) [2 marks]

What is the total number of **primary rays** shot?

- **A.** 10,000
- **B.** 50,000
- **C.** 60,000 **(answer)**
- **D.** 70,000
- **E.** 100,000
- **F.** 120,000
- **G.** 140,000

(15) [2 marks]

What is the total number of **shadow rays** shot from those points intersected by the primary rays?

- **A.** 0
- **B.** 20,000
- **C.** 100,000
- **D.** 120,000 **(answer)**
- **E.** 140,000
- **F.** More than 20,000 and less than 100,000
- **G.** More than 100,000 and less than 120,000
- **H.** More than 120,000 and less than 140,000

(16) [3 marks]

How many **first-level** secondary **reflection rays** are spawned? First-level secondary reflection rays are spawned from surface points hit by the primary rays.

- **A.** 0
- **B.** 10,000 **(answer)**
- **C.** 50,000
- **D.** 60,000
- **E.** 70,000
- **F.** 100,000
- **G.** 120,000
- **H.** 140,000

(17) [3 marks]

How many **second-level** secondary **reflection rays** are spawned? Second-level secondary reflection rays are spawned from surface points hit by the first-level secondary reflection rays.

- **A.** 0 **(answer)**
- **B.** 10,000
- **C.** 50,000
- **D.** 60,000
- **E.** 70,000
- **F.** 100,000
- **G.** 120,000
- **H.** 140,000

(18) [3 marks]

How many **rays** are shot **altogether**?

10,000 * 6 + 50,000 * 3 = 210,000

Section G [6 marks]

(19) [4 marks]

Consider the **ray** (in parametric representation)

$$\boldsymbol{P}(t) = \begin{bmatrix} 16 \\ 4 \\ 9 \end{bmatrix} + t \begin{bmatrix} -8 \\ 1 \\ 1 \end{bmatrix}$$

and the **sphere** ***S*** (in implicit representation)

$$(x-8)^2 + (y-5)^2 + (z-10)^2 - 6^2 = 0$$

Suppose we are only concerned with finding the value of *t* at the first intersection between the ray $\boldsymbol{P}(t)$ and the sphere $\boldsymbol{S}$. This is equivalent to finding the value of *t* at the first intersection between the ray $\boldsymbol{R}(t)$ and the sphere

$$x^2 + y^2 + z^2 - 6^2 = 0$$

Which of the followings is the correct $\boldsymbol{R}(t)$?

A. $\boldsymbol{R}(t) = [0 \;\; 0 \;\; 0]^T + t\,[-8 \;\; 1 \;\; 1]^T$
B. $\boldsymbol{R}(t) = [16 \;\; 4 \;\; 9]^T + t\,[-8 \;\; 1 \;\; 1]^T$
C. $\boldsymbol{R}(t) = [8 \;\; 5 \;\; 10]^T + t\,[-8 \;\; 1 \;\; 1]^T$
D. $\boldsymbol{R}(t) = [24 \;\; 9 \;\; 19]^T + t\,[-8 \;\; 1 \;\; 1]^T$
E. $\boldsymbol{R}(t) = [8 \;\; -1 \;\; -1]^T + t\,[-8 \;\; 1 \;\; 1]^T$ **(answer)**
F. $\boldsymbol{R}(t) = [-8 \;\; 1 \;\; 1]^T + t\,[-8 \;\; 1 \;\; 1]^T$

(20) [2 marks]

In Whitted Ray Tracing, when a ray $\boldsymbol{P}(t) = \boldsymbol{O} + t\,\boldsymbol{D}$ is being intersected with an **opaque sphere**, we often consider only the intersection at the smaller *t* value. In what situation should we instead consider the intersection at the larger *t* value?

When the light ray's origin is inside the sphere.
Or, the whole scene is surrounded by the sphere.

Section H [23 marks]

(21) [3 marks]

Consider the following **explicit form** representation of an infinite **2D curve**:

$$y = 4x^5 + \sin^3 6x$$

Write the **parametric form** representation of the curve, parameterized by the independent variable u.

You must write your answer in the following form:

$x(u)$ = *expression in terms of u*
$y(u)$ = *expression in terms of u*

$$x(u) = u$$

$$y(u) = 4u^5 + sin^3 6u$$

(22) [4 marks]

Consider the following **parametric cubic polynomial 2D curve segment** for $0 \le u \le 1$:

$$\mathbf{p}(u) = \begin{bmatrix}1\\4\end{bmatrix} + \begin{bmatrix}6\\6\end{bmatrix}u + \begin{bmatrix}9\\27\end{bmatrix}u^2 + \begin{bmatrix}27\\54\end{bmatrix}u^3$$

Write the coordinates of the **four control points**, $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$, for the **cubic interpolating curve segment** that corresponds to $\mathbf{p}(u)$. The control points must be given in the right order such that $\mathbf{p}_0$ corresponds to $\mathbf{p}(0)$, and so on.

You must write your answer in the following form:

$\mathbf{p}_0$ = (*x-coordinate*, *y-coordinate*)
$\mathbf{p}_1$ = (*x-coordinate*, *y-coordinate*)
$\mathbf{p}_2$ = (*x-coordinate*, *y-coordinate*)
$\mathbf{p}_3$ = (*x-coordinate*, *y-coordinate*)

$\mathbf{p}_0 = \mathbf{p}(0) = (1,\ 4)$
$\mathbf{p}_1 = \mathbf{p}(1/3) = (5,\ 11)$
$\mathbf{p}_2 = \mathbf{p}(2/3) = (17,\ 36)$
$\mathbf{p}_3 = \mathbf{p}(1) = (43,\ 91)$

(23) [5 marks]

Consider the following **parametric cubic polynomial 2D curve segment** for $0 \le u \le 1$:

$$\mathbf{p}(u) = \begin{bmatrix}1\\2\end{bmatrix} + \begin{bmatrix}3\\9\end{bmatrix} u + \begin{bmatrix}3\\3\end{bmatrix} u^2 + \begin{bmatrix}3\\6\end{bmatrix} u^3$$

Write the coordinates of the **four control points**, $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$, for the **cubic Bézier curve segment** that corresponds to $\mathbf{p}(u)$. The control points must be given in the right order such that $\mathbf{p}_0$ corresponds to $\mathbf{p}(0)$, and so on.

You must write your answer in the following form:

$\mathbf{p}_0$ = (*x-coordinate*, *y-coordinate*)
$\mathbf{p}_1$ = (*x-coordinate*, *y-coordinate*)
$\mathbf{p}_2$ = (*x-coordinate*, *y-coordinate*)
$\mathbf{p}_3$ = (*x-coordinate*, *y-coordinate*)

$\mathbf{p}_0 = \mathbf{p}(0)$ = (1, 2) —— [0.5 marks]
$\mathbf{p}_1 = (\mathbf{p}'(0) + 3\mathbf{p}_0) / 3$ = (3, 9) / 3 + (1, 2) = (2, 5) —— [2 marks]
$\mathbf{p}_2 = (3\mathbf{p}_3 - \mathbf{p}'(1)) / 3$ = (10, 20) – (18, 33) / 3 = (4, 9) —— [2 marks]
$\mathbf{p}_3 = \mathbf{p}(1)$ = (10, 20) —— [0.5 marks]

(24) [6 marks]

Consider the following **two parametric cubic polynomial 2D curve segments** for $0 \le u \le 1$:

$$\mathbf{p}(u) = (1-u)^3 \begin{bmatrix}1\\10\end{bmatrix} + 3u(1-u)^2 \begin{bmatrix}2\\11\end{bmatrix} + 3u^2(1-u) \begin{bmatrix}4\\11\end{bmatrix} + u^3 \begin{bmatrix}5\\10\end{bmatrix}$$

$$\mathbf{q}(u) = (1-u)^3 \begin{bmatrix}5\\10\end{bmatrix} + 3u(1-u)^2 \begin{bmatrix}8\\7\end{bmatrix} + 3u^2(1-u) \begin{bmatrix}10\\7\end{bmatrix} + u^3 \begin{bmatrix}15\\5\end{bmatrix}$$

If the two curve segments are joined together at $\mathbf{p}(1)$ and $\mathbf{q}(0)$, for each of the following statements, indicate it is true or false.

(i) $\mathbf{p}(u)$ and $\mathbf{q}(u)$ have C^0 continuity at their joint point. (True or False?)

(ii) $\mathbf{p}(u)$ and $\mathbf{q}(u)$ have C^1 continuity at their joint point. (True or False?)

(iii) $\mathbf{p}(u)$ and $\mathbf{q}(u)$ have G^1 continuity at their joint point. (True or False?)

You must write your answer in the following form:

(i) True *or* False
(ii) True *or* False
(iii) True *or* False

(i) True
(ii) False
(iii) True

(25) [5 marks]

Consider the **cubic Bézier 2D curve segment** $\mathbf{p}(u)$, for $0 \leq u \leq 1$, whose control points are

$\mathbf{p}_0 = (2, 6)$
$\mathbf{p}_1 = (0, 12)$
$\mathbf{p}_2 = (10, 10)$
$\mathbf{p}_3 = (16, 0)$

Suppose $\mathbf{p}(u)$ is split into two segments at $u = 0.5$, where the sub-segment $\mathbf{s}(t)$, for $0 \leq t \leq 1$, is the same as $\mathbf{p}(u)$ for $0 \leq u \leq 0.5$.

Write the coordinates of the **four control points**, $\mathbf{s}_0$, $\mathbf{s}_1$, $\mathbf{s}_2$ and $\mathbf{s}_3$, for the **cubic Bézier curve segment** that corresponds to $\mathbf{s}(t)$. The control points must be given in the right order such that $\mathbf{s}_0$ corresponds to $\mathbf{s}(0)$, and so on.

You must write your answer in the following form:

$\mathbf{s}_0$ = (*x-coordinate*, *y-coordinate*)
$\mathbf{s}_1$ = (*x-coordinate*, *y-coordinate*)
$\mathbf{s}_2$ = (*x-coordinate*, *y-coordinate*)
$\mathbf{s}_3$ = (*x-coordinate*, *y-coordinate*)

$\mathbf{s}_0 = \mathbf{p}(0) = (2,\ 6)$ —— [0 marks]
$\mathbf{s}_1 = (1,\ 9)$ —— [2 marks]
$\mathbf{s}_2 = (3,\ 10)$ —— [2 marks]
$\mathbf{s}_3 = \mathbf{p}(0.5) = (6,\ 9)$ —— [1 marks]

—— END OF PAPER ——