NATIONAL UNIVERSITY OF SINGAPORE

CS3241 — COMPUTER GRAPHICS

AY2022/2023 SEMESTER 1

MIDTERM ASSESSMENT

Time Allowed: **1 Hour 30 Minutes**

INSTRUCTIONS

1. This is an **OPEN BOOK** assessment.

2. This question paper comprises **TEN (10)** printed pages, including this page.

3. You are also provided with **one OCR Form** to write your answers.

4. Clearly **write** and **shade** your **STUDENT NUMBER** in **SECTION B** of your **OCR Form** using a **2B PENCIL**.

5. You do not need to write in **SECTION A** of your OCR Form.

6. There are **31** multiple-choice questions (MCQs) in **6** Sections. Each question has one correct answer. The **indicated marks** are awarded for each correct answer and there is **no penalty** for a wrong answer.

7. The full score is **80 marks**.

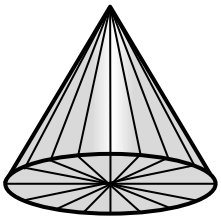
8. Answer **ALL** questions.

9. Use only a **2B PENCIL** to **shade** your answers on your **OCR Form**.

10. **Submit only the OCR Form**.

Section A [8 marks]

A closed cone is approximated and represented as a set of triangles as shown in the diagram, where the top curved surface is approximated by 16 triangles, and the flat bottom surface is also approximated by 16 triangles.



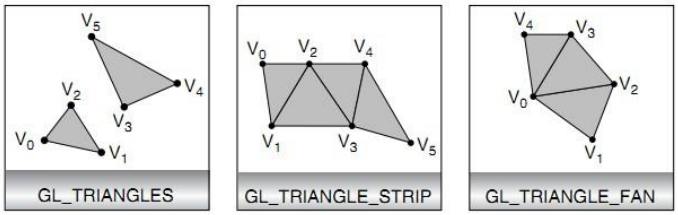
(1) **[2 marks]** How many ***distinct*** **vertices** are there in the representation of the cone?

A. 10
B. 16
C. 18
D. 34
E. 96

(2) **[3 marks]** Suppose we want to draw the cone using the `GL_TRIANGLES` OpenGL primitive mode, how many times do we need to call the `glVertex*()` function?

A. 32
B. 34
C. 64
D. 66
E. 96

(3) **[3 marks]** We want to draw the entire cone as two `GL_TRIANGLE_FAN` primitives. In total, how many times do we need to call the `glVertex*()` function for drawing the entire cone?



A. 32
B. 33
C. 34
D. 36
E. 66

Section B [12 marks]

(4) **[3 marks]** Suppose we have 3 pieces of glass filters of equal thickness, of which one is **cyan** color, one is **magenta**, and one is **yellow**. We want to shine a beam of white light through the glass filters to produce **green light**. Which is/are the glass filter(s) that we should shine the white light through?

A. Cyan, magenta and yellow
B. Cyan and magenta
C. Magenta and yellow
D. Yellow and cyan
E. Magenta only

(5) **[3 marks]** A beam of white light is shining through an alternating glass filter that is changing color between **cyan** and **magenta** at a rate of 1000 Hz. Suppose the filtered light beam falls on a white matte opaque surface, what will be the perceived color on the white surface?

A. Light blue
B. Dark blue
C. Green
D. Red
E. Yellow

(6) **[3 marks]** How much framebuffer memory is required for a 600×480 window, with 16-bit color (i.e. for RGBA combined), 32-bit depth value and using **single-buffering**?

A. $600 \times 480 \times (2 + 4)$ bytes
B. $600 \times 480 \times (2 \times 2 + 4)$ bytes
C. $600 \times 480 \times (2 + 4) \times 2$ bytes
D. $600 \times 480 \times (16 + 32)$ bytes
E. $600 \times 480 \times (16 + 32) \times 2$ bytes

(7) **[3 marks]** How much framebuffer memory is required for a 600×480 window, with 16-bit color (i.e. for RGBA combined), 32-bit depth value and using **double-buffering**?

A. $600 \times 480 \times (2 + 4)$ bytes
B. $600 \times 480 \times (2 \times 2 + 4)$ bytes
C. $600 \times 480 \times (2 + 4) \times 2$ bytes
D. $600 \times 480 \times (16 + 32)$ bytes
E. $600 \times 480 \times (16 + 32) \times 2$ bytes

Section C [21 marks]

(8) **[3 marks]** What does the homogeneous coordinates $[9\ \ 12\ \ 6\ \ 3]^T$ represent?

A. The 3D point (3, 4, 2)
B. The 3D point (9, 12, 6)
C. The 3D point (27, 36, 18)
D. The 3D vector (3, 4, 2)
E. The 3D vector (27, 36, 18)

(9) **[3 marks]** Which of the following is the matrix that rotates objects about the point (6, 4, 8), where the rotation axis is the vector (0, 1, 0), and the rotation angle is θ? Note that $\mathbf{T}(d_x, d_y, d_z)$ is a translation matrix for displacing a point by (d_x, d_y, d_z), and $\mathbf{R}_y(\alpha)$ is a rotation matrix for rotating a point about the *y*-axis by an angle of α.

A. $\mathbf{T}(-6, -4, -8) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{T}(6, 4, 8)$
B. $\mathbf{T}(-6, -4, -8) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{T}(-6, -4, -8)$
C. $\mathbf{T}(6, 4, 8) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{T}(6, 4, 8)$
D. $\mathbf{T}(6, -4, 8) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{T}(-6, 4, -8)$
E. $\mathbf{T}(-6, 4, -8) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{T}(6, -4, 8)$

(10) **[3 marks]** A vertex, whose **camera coordinates** are (50, 50, –25), is being projected using the following OpenGL orthographic projection:

```
glOrtho( -100, 100, -200, 200, 0, 100 );
```

What will be the vertex's Normalized Device Coordinates (NDC)?

A. (0.5, 0.25, –0.5)
B. (0.5, 0.25, 0.5)
C. (0.5, 0.25, –0.25)
D. (0.5, 0.25, 0.25)
E. (0,75, 0.625, 0.25)

(11) **[3 marks]** Suppose the **viewport** is set up as `glViewport(200, 100, 600, 400)`, and the entire viewport is within the rendering window, what is the **2D window coordinates** and **depth value** of the NDC point (1, 1, 1)?

A. Window coordinates: (200, 100), and depth: 0.0
B. Window coordinates: (600, 400), and depth: 0.0
C. Window coordinates: (600, 400), and depth: 1.0
D. Window coordinates: (800, 500), and depth: 0.0
E. Window coordinates: (800, 500), and depth: 1.0

(12) **[3 marks]** Suppose two diagonally-opposite corners of an **orthographic viewing volume** are at (–40, –30, –100) and (10, 50, –60) in the camera space, which of the following `glOrtho` function calls correctly sets up the viewing volume?

A. `glOrtho(-10, 40, -50, 30, 60, 100);`
B. `glOrtho(-40, 10, -30, 50, 60, 100);`
C. `glOrtho(-40, 10, -30, 50, -60, -100);`
D. `glOrtho(-40, 10, -30, 50, 100, 60);`
E. `glOrtho(-40, 10, -30, 50, -100, -60);`

(13) **[3 marks]** Suppose two diagonally-opposite corners on the far plane of a **perspective viewing volume** are at (–40, –30, –100) and (10, 50, –100) in the camera space, and the near plane distance is 50, which of the following `glFrustum` function calls correctly sets up the viewing volume?

A. `glFrustum(-40, 10, -30, 50, 50, 100);`
B. `glFrustum(-40, 10, -30, 50, -50, -100);`
C. `glFrustum(-20, 5, -15, 25, 50, 100);`
D. `glFrustum(-20, 5, -15, 25, -50, -100);`
E. `glFrustum(-10, 40, -50, 30, 50, 100);`

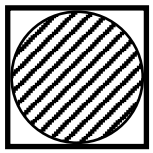
(14) **[3 marks]** Which of the following is equivalent to this `gluLookAt()` function call:

```
gluLookAt( ex, ey, ez,  ex+1, ey, ez-1,  1, 1, -1 );
```

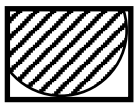
A. `glRotated(90, 0, 1, 0); glTranslated(-ex, -ey, -ez);`
B. `glTranslated(-ex, -ey, -ez); glRotated(90, 0, 1, 0);`
C. `glRotated(45, 0, 0, 1); glTranslated(-ex, -ey, -ez);`
D. `glTranslated(-ex, -ey, -ez); glRotated(45, 0, 1, 0);`
E. `glRotated(45, 0, 1, 0); glTranslated(-ex, -ey, -ez);`

Section D [12 marks]

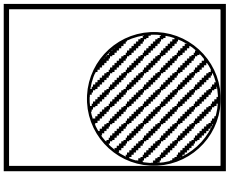
Suppose in the **camera coordinate frame**, there is a disc in the $z = 0$ plane, **centered** at (200, 300, 0), and has a **radius** of **50**. Initially, the disc fits exactly in a window of size 600×600. When the window is resized, you want the disc to appear the same size on screen as before, even though it may be clipped; or if the window is too large, the disc always appears in the bottom-right corner (see diagram below). Complete the **reshape callback** function, in the next 4 questions, to set up the required **orthographic projection**. The **viewport** is always set to be the whole window. Function usage: `gluOrtho2D(left, right, bottom, top);`



Initial window



Shrunk window



Enlarged window

```
void MyReshape( int w, int h ) {

    glViewport( 0, 0, w, h );

    glMatrixMode( GL_PROJECTION );
    glLoadIdentity();

    float left = ???
    float right = ???
    float bottom = ???
    float top = ???

    gluOrtho2D( left, right, bottom, top );

    glMatrixMode( GL_MODELVIEW );
}
```

(15) [3 marks] What should be the value for the `left` argument passed to `gluOrtho2D`?

A. `float left = 200 + 50 – (100 * (float)w/600);`
B. `float left = 200 + 50 + (100 * (float)w/600);`
C. `float left = 200 - 50 + (100 * (float)w/600);`
D. `float left = 200 + 50;`
E. `float left = 200 - 50;`

(16) **[3 marks]** What should be the value for the `right` argument passed to `gluOrtho2D`?

A. `float right = 200 + 50 – (100 * (float)w/600);`
B. `float right = 200 + 50 + (100 * (float)w/600);`
C. `float right = 200 - 50 + (100 * (float)w/600);`
D. `float right = 200 + 50;`
E. `float right = 200 - 50;`

(17) **[3 marks]** What should be the value for the `bottom` argument passed to `gluOrtho2D`?

A. `float bottom = 300 + 50 – (100 * (float)h/600);`
B. `float bottom = 300 + 50 + (100 * (float)h/600);`
C. `float bottom = 300 - 50 + (100 * (float)h/600);`
D. `float bottom = 300 + 50;`
E. `float bottom = 300 - 50;`

(18) **[3 marks]** What should be the value for the `top` argument passed to `gluOrtho2D`?

A. `float top = 300 + 50 – (100 * (float)h/600);`
B. `float top = 300 + 50 + (100 * (float)h/600);`
C. `float top = 300 - 50 + (100 * (float)h/600);`
D. `float top = 300 + 50;`
E. `float top = 300 - 50;`

Section E [6 marks]

For each of the following statements, indicate whether it is **true** or **false**.

(19) **[1 mark]** Assigning a RGBA color, where the A value is 0.5, to all the vertices of a polygon, will automatically make the polygon look translucent. True or false?

A. True
B. False

(20) **[1 mark]** The fragment processing stage can *directly* read the content of the z-buffer to use it for depth testing (*z* testing). True or false?

A. True
B. False

(21) **[1 mark]** After a fragment is processed by the per-fragment operations stage, it can be written to only one pixel location in the color buffer. True or false?

A. True
B. False

(22) **[1 mark]** Color values on a polygon are not interpolated (i.e. constant across the polygon) when flat shading is turned on. True or false?

A. True
B. False

(23) **[1 mark]** Depth values (z values) on a polygon are not interpolated (i.e. constant across the polygon) when flat shading is turned on. True or false?

A. True
B. False

(24) **[1 mark]** Depth testing (z testing) on a fragment is performed after texture mapping in the OpenGL rendering pipeline. True or false?

A. True
B. False

Section F [21 marks]

(25) [3 marks] Pixel P in the framebuffer is covered by triangles T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, and T_{10} only. The triangles are rendered in the order T_1, T_2, …, T_{10}. Suppose the **depth values** of the fragments of T_1, T_2, …, T_{10} at the location of P are 0.95, 0.85, 0.50, 0.65, 0.20, 0.90, 0.15, 0.30, 0.70, 0.10, respectively, how many times will the z-value in the **z-buffer** for pixel P be modified *after* it was initialized?

A. 7
B. 6
C. 5
D. 4
E. 3

(26) [2 marks] Consider using the Cohen-Sutherland Algorithm to clip a **2D line segment** against a rectangular clipping window. Suppose the two endpoints have outcodes `1010` and `0100`, what is the maximum number of **intersections** that need to be computed?

A. 0
B. 1
C. 2
D. 3
E. 4

(27) [2 marks] Consider using the Cohen-Sutherland Algorithm to clip a **2D line segment** against a rectangular clipping window. Suppose the two endpoints have outcodes `1010` and `0010`, what is the maximum number of **intersections** that need to be computed?

A. 0
B. 1
C. 2
D. 3
E. 4

(28) [2 marks] Consider using the Cohen-Sutherland Algorithm to clip a **2D line segment** against a rectangular clipping window. Suppose the two endpoints have outcodes `1001` and `0110`, what is the maximum number of **intersections** that need to be computed?

A. 0
B. 1
C. 2
D. 3
E. 4

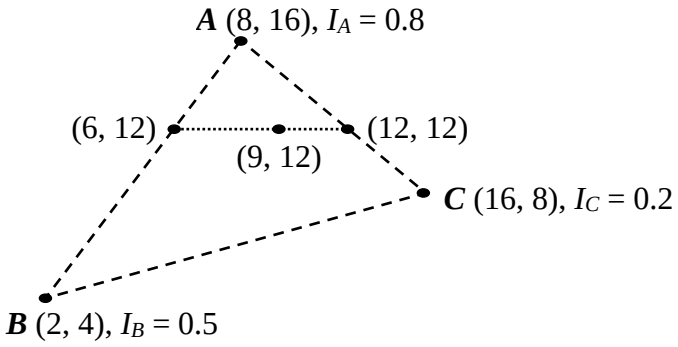
(29) **[4 marks]** **Two** straight **line segments** are being **scan-converted**. The first line segment has endpoints at pixel locations $\boldsymbol{A}_1(31, 31)$ and $\boldsymbol{A}_2(60, 80)$, and the second line segment has endpoints at pixel locations $\boldsymbol{B}_1(1, 1)$ and $\boldsymbol{B}_2(105, 85)$. Assuming the line segments are drawn as thin as possible and not broken, what is the **total number of fragments** (pixels that are turned on) that are produced for these two line segments? This number includes the four fragments at the four end points of the two line segments.

A. 50 + 105
B. 30 + 105
C. 50 + 85
D. 30 + 85
E. 85 + 105

(30) **[4 marks]** A curve of $y = (4x^2 + 128) / 64$ is scan-converted from the pixel location (0, 2) to (52, 171). Assuming the curve is drawn as thin as possible and not broken, what is the **number of fragments** (pixels that are turned on) that are produced for this curve? This number includes the two fragments at the two end points of the curve.

A. 53
B. 170
C. 174
D. 180
E. 222

(31) **[4 marks]** Given the following triangle whose vertices ***A***, ***B***, ***C*** are at *pixel locations* (8, 16), (2, 4), and (16, 8) respectively, and the **intensity** attribute values at the vertices are 0.8, 0.5, 0.2 respectively. (Note that the diagram is not drawn to scale.) What is the intensity value at the pixel location (9, 12) assuming the triangle is rasterized with smooth shading?



A. 0.45
B. 0.50
C. 0.55
D. 0.60
E. 0.65

——— END OF QUESTIONS ———