NATIONAL UNIVERSITY OF SINGAPORE

CS3241 — COMPUTER GRAPHICS

(AY2021/2022 SEMESTER 1)

MIDTERM ASSESSMENT

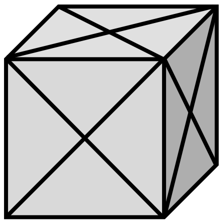
Time Allowed: **1 Hour 30 Minutes**

INSTRUCTIONS

1. This assessment contains **28 Questions** in **7 Sections**.
2. The full score of this assessment is **72 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 [12 marks]

A cube is represented as a set of triangles as shown in the following diagram. All six faces of the cube are represented the same way. For the questions in this section, consider this polygonal representation of the cube.



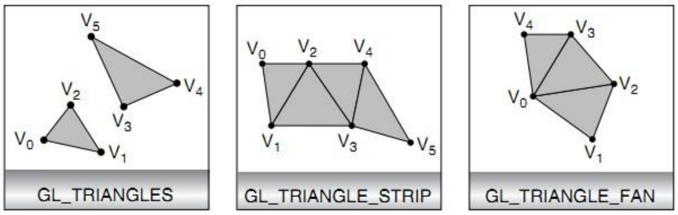
(1) **[2 marks]** How many ***distinct*** **vertices** are there in the representation of the cube?

- **A.** 6
- **B.** 8
- **C.** 12
- **D.** 14
- **E.** 16
- **F.** 24

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

- **A.** 18
- **B.** 24
- **C.** 36
- **D.** 48
- **E.** 72
- **F.** 96

(3) **[2 marks]** We want to draw **each face** of the cube as a `GL_TRIANGLE_FAN` primitive. In total, how many times do we need to call the `glVertex*()` function for drawing the entire cube?



- **A.** 18
- **B.** 24
- **C.** 30
- **D.** 36
- **E.** 42
- **F.** 54

(4) **[3 marks]** Suppose the cube is rendered with a **perspective projection**, and it is entirely within the view volume, what would be the **minimum** number of **triangles** that could be eliminated by **back-face culling**?

- **A.** 2
- **B.** 3
- **C.** 4
- **D.** 5
- **E.** 8
- **F.** 12
- **G.** 16
- **H.** 20

(5) **[3 marks]** Suppose a particular implementation of **back-face culling** uses the **window coordinates** of the vertices of polygons. Which of the following OpenGL pipeline operations will have the most significant **reduction** in **workload** as a result of this back-face culling?

- **A.** Rasterization
- **B.** Primitive assembly
- **C.** Viewport transformation
- **D.** Clipping
- **E.** View transformation
- **F.** Perspective division
- **G.** Vertex processing

Section B [8 marks]

(6) **[2 marks]** Given the R, G and B color values of a pixel, we would like to compute the **perceived brightness** of the pixel using the formula $k_R \cdot R + k_G \cdot G + k_B \cdot B$, where a higher value corresponds to a higher brightness. Which of the following are the **most suitable** values for constants k_R, k_G, k_B, respectively?

- **A.** $k_R = 0.2$, $k_G = 0.3$, $k_B = 0.5$
- **B.** $k_R = 0.5$, $k_G = 0.1$, $k_B = 0.4$
- **C.** $k_R = 0.3$, $k_G = 0.4$, $k_B = 0.3$
- **D.** $k_R = 0.2$, $k_G = 0.4$, $k_B = 0.4$
- **E.** $k_R = 0.3$, $k_G = 0.6$, $k_B = 0.1$
- **F.** $k_R = 0.1$, $k_G = 0.6$, $k_B = 0.3$

(7) **[3 marks]** Suppose an inkjet printer has used up its **magenta** ink, and it is left with only **cyan** and **yellow** ink. Which of the following colors **cannot** be produced by the printer on a piece of white paper?

- **A.** Red and green
- **B.** Red and blue
- **C.** Green and blue
- **D.** Green and magenta
- **E.** Cyan and yellow
- **F.** White

(8) **[3 marks]** Let F_1 be a flat piece of colored glass that lets 50% of red light, 50% of green light, and 90% of blue light, pass through. Another flat piece of glass, F_2, lets 40% of red light, 20% of green light, and 10% of blue light, pass through. If we shine a beam of white light through glass F_1 and glass F_2, in that order, how much of the white light actually passes through both F_1 and F_2?

- **A.** 20% of red, 10% of green, 9% of blue
- **B.** 10% of red, 30% of green, 80% of blue
- **C.** 90% of red, 70% of green, 100% of blue
- **D.** 50% of red, 50% of green, 90% of blue
- **E.** 40% of red, 20% of green, 10% of blue
- **F.** 50% of red, 50% of green, 50% of blue

Section C [12 marks]

(9) **[1 mark]** Given that 3D vectors *U* and *V* are **unit vectors**, what is the value of their **dot product** (i.e. $U \bullet V$) when *U* and *V* are **perpendicular** to each other?

- **A.** –1
- **B.** 0
- **C.** 1
- **D.** $\pi / 2$
- **E.** $\pi / 4$
- **F.** A value greater than –1 and less than 0
- **G.** A value greater than 0 and less than 1
- **H.** The result is a 3D vector
- **I.** None of the other options is the correct answer

(10) **[1 mark]** Given that 3D vectors *U* and *V* are **unit vectors**, what is the value of their **dot product** (i.e. $U \bullet V$) when the angle between *U* and *V* is larger than a right angle and smaller than 180 degrees?

- **A.** –1
- **B.** 0
- **C.** 1
- **D.** $\pi / 2$
- **E.** $\pi / 4$
- **F.** A value greater than –1 and less than 0
- **G.** A value greater than 0 and less than 1
- **H.** The result is a 3D vector
- **I.** None of the other options is the correct answer

(11) **[2 marks]** Given that 3D vectors *U* and *V* are **unit vectors** that are not parallel to each other, which of the following is correct about the **cross product** of *U* and *V* (i.e. $U \times V$)?

- **A.** $U \times V$ is a scalar value
- **B.** $U \times V$ is a 3D point
- **C.** $U \times V$ is a 3D vector that is perpendicular to *U* but not perpendicular to *V*
- **D.** $U \times V$ is a 3D vector that is perpendicular to *V* but not perpendicular to *U*
- **E.** $U \times V$ is a 3D vector that is perpendicular to both *U* and *V*
- **F.** $U \times V$ is a 3D vector that is parallel to *U* but not parallel to *V*
- **G.** $U \times V$ is a 3D vector that is parallel to *V* but not parallel to *U*
- **H.** $U \times V$ is a 3D vector that is parallel to both *U* and *V*

(12) **[2 marks]** What does the homogeneous coordinates $[2\ \ 3\ \ 1\ \ 5]^T$ represent?

- **A.** The 3D point (2, 3, 1)
- **B.** The 3D point (10, 15, 5)
- **C.** The 3D point (2/5, 3/5, 1/5)
- **D.** The 3D point (5/2, 5/3, 5/1)
- **E.** The 3D vector (2, 3, 1)
- **F.** The 3D vector (10, 15, 5)
- **G.** The 3D vector (2/5, 3/5, 1/5)
- **H.** The 3D vector (5/2, 5/3, 5/1)

(13) **[3 marks]** Given homogeneous coordinates $[3\ \ 12\ \ 9\ \ 3]^T$ and $[1\ \ 2\ \ 3\ \ 0]^T$, what does $[3\ \ 12\ \ 9\ \ 3]^T + [1\ \ 2\ \ 3\ \ 0]^T$ represent?

- **A.** The 3D point (4, 14, 12)
- **B.** The 3D point (2, 6, 6)
- **C.** The 3D point (4/3, 14/3, 4)
- **D.** The 3D point (3/4, 3/14/, 1/4)
- **E.** The 3D vector (4, 14, 12)
- **F.** The 3D vector (2, 6, 6)
- **G.** The 3D vector (4/3, 14/3, 4)
- **H.** The 3D vector (3/4, 3/14/, 1/4)

(14) **[3 marks]** Which of the following is the matrix that rotates objects about the point (3, 8, 5), where the rotation axis is the vector (1, 0, 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 $\mathbf{R}_x(\alpha)$ is a rotation matrix for rotating a point about the x-axis by an angle of α.

- **A.** $\mathbf{T}(-3, -8, 5) \cdot \mathbf{R}_x(\theta) \cdot \mathbf{T}(3, 8, -5)$
- **B.** $\mathbf{T}(-3, 8, -5) \cdot \mathbf{R}_x(\theta) \cdot \mathbf{T}(3, -8, 5)$
- **C.** $\mathbf{T}(3, -8, -5) \cdot \mathbf{R}_x(\theta) \cdot \mathbf{T}(-3, 8, 5)$
- **D.** $\mathbf{T}(3, 8, -5) \cdot \mathbf{R}_x(\theta) \cdot \mathbf{T}(-3, -8, 5)$
- **E.** $\mathbf{T}(3, -8, 5) \cdot \mathbf{R}_x(\theta) \cdot \mathbf{T}(-3, 8, -5)$
- **F.** $\mathbf{T}(-3, 8, 5) \cdot \mathbf{R}_x(\theta) \cdot \mathbf{T}(3, -8, -5)$

Section D [10 marks]

(15) **[4 marks]** Suppose we want to position the camera at the world-space location (`ex`, `ey`, `ez`), with the camera's view direction pointing in the *y* direction, and the "up-vector" pointing in the direction (1, 0, 1). Which of the following `gluLookAt()` function calls achieves an equivalent view transformation?

- **A.** `gluLookAt( ex, ey, ez,  0, 1, 0,  1, 0, 1 );`
- **B.** `gluLookAt( ex, ey, ez,  0, -1, 0,  -1, 0, -1 );`
- **C.** `gluLookAt( ex, ey, ez,  ex, ey+5, ez,  3, 1, 3 );`
- **D.** `gluLookAt( ex, ey, ez,  ex, ey-5, ez,  -1, 1, -1 );`
- **E.** `gluLookAt( ex, ey, ez,  ex, ey+1, ez,  -1, 0, -1 );`
- **F.** `gluLookAt( ex, ey, ez,  ex, ey-1, ez,   1, 0, 1 );`
- **G.** None of the other options is the correct answer

(16) **[3 marks]** Consider the following OpenGL `glOrtho()` function call:

```
glOrtho( 0, 10, 0, 20, 0, 100 );
```

Which of the following transformation matrices is the one produced by the `glOrtho()` function call? 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 $\mathbf{S}(s_x, s_y, s_z)$ is a scaling matrix for scaling a point by factors of s_x, s_y, s_z in the x, y, z directions, respectively.

- **A.** $\mathbf{S}(2/10, 2/20, -2/100) \cdot \mathbf{T}(-5, -10, 50)$
- **B.** $\mathbf{S}(2/10, 2/20, -2/100) \cdot \mathbf{T}(-5, -10, -50)$
- **C.** $\mathbf{S}(2/10, 2/20, 2/100) \cdot \mathbf{T}(-5, -10, 50)$
- **D.** $\mathbf{S}(2/10, 2/20, 2/100) \cdot \mathbf{T}(-5, -10, -50)$
- **E.** $\mathbf{T}(-5, -10, 50) \cdot \mathbf{S}(2/10, 2/20, -2/100)$
- **F.** $\mathbf{T}(-5, -10, -50) \cdot \mathbf{S}(2/10, 2/20, -2/100)$

(17) **[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 (0, 0, 0)?

- **A.** Window coordinates: (400, 250), and depth: 1.0
- **B.** Window coordinates: (400, 250), and depth: 0.0
- **C.** Window coordinates: (400, 250), and depth: 0.5
- **D.** Window coordinates: (500, 300), and depth: 1.0
- **E.** Window coordinates: (500, 300), and depth: 0.0
- **F.** Window coordinates: (500, 300), and depth: 0.5

Section E [9 marks]

For the questions in this section, consider the following application program code fragment:

```
glMatrixMode( GL_MODELVIEW );
glLoadMatrixd( A );
glPushMatrix();
    glMultMatrixd( B );
    glBegin( GL_POINTS ); glVertex3dv( v1 ); glEnd();
    glMultMatrixd( C );
    glLoadMatrixd( D );
    glPushMatrix();
        glMultMatrixd( E );
        glMultMatrixd( F );
        glBegin( GL_POINTS ); glVertex3dv( v2 ); glEnd();
        glLoadMatrixd( G );
    glPopMatrix();
    glMultMatrixd( H );
    glBegin( GL_POINTS ); glVertex3dv( v3 ); glEnd();
    glPushMatrix();
        glMultMatrixd( J );
        glPushMatrix();
            glMultMatrixd( K );
            glBegin( GL_POINTS ); glVertex3dv( v4 ); glEnd();
        glPopMatrix();
        glMultMatrixd( L );
        glBegin( GL_POINTS ); glVertex3dv( v5 ); glEnd();
    glPopMatrix();
glPopMatrix();
```

(18) [3 marks] What is the actual transformations applied to the vertex **v1**?

- **A.** **A·v1**
- **B.** **B·v1**
- **C.** **A·B·v1**
- **D.** **B·A·v1**
- **E.** **C·D·v1**
- **F.** **D·C·v1**
- **G.** **A·B·C·D·v1**
- **H.** None of the other options is the correct answer

(19) **[3 marks]** What is the actual transformations applied to the vertex **v2**?

A. **A·B·C·D·E·F·v2**
B. **F·E·D·C·B·A·v2**
C. **B·C·D·E·F·v2**
D. **F·E·D·C·B·v2**
E. **D·E·F·C·v2·G**
F. **D·E·F·v2**
G. **F·E·D·v2**
H. None of the other options is the correct answer

(20) **[3 marks]** What is the actual transformations applied to the vertex **v3**?

A. **D·H·v3**
B. **G·H·v3**
C. **G·D·H·v3**
D. **D·G·H·v3**
E. **A·B·C·D·H·v3**
F. **D·E·F·G·H·v3**
G. **A·B·C·D·E·F·G·H·v3**
H. None of the other options is the correct answer

Section F [9 marks]

Consider the following **view-transformation matrix**.

$$\boldsymbol{V} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & m \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(21) **[3 marks]** What kind of transformation is represented by the top-left 3-by-3 submatrix of ***V*** ?

- **A.** Translation
- **B.** Uniform scaling (i.e. same scaling factor for *x*, *y*, *z* coordinates)
- **C.** Non-uniform scaling
- **D.** Rotation
- **E.** Shear
- **F.** Orthographic projection
- **G.** Perspective projection

(22) **[3 marks]** Given the above **view-transformation** matrix ***V***, which of the following is the matrix that can be used to transform a **3D vector** from **world space** to **view space**?

A. $\begin{bmatrix} a & b & c \\ e & f & g \\ i & j & k \end{bmatrix}$

B. $\begin{bmatrix} a & e & i \\ b & f & j \\ c & g & k \end{bmatrix}$

C. $\begin{bmatrix} 1/a & b & c \\ e & 1/f & g \\ i & j & 1/k \end{bmatrix}$

D. $\begin{bmatrix} 1/a & e & i \\ b & 1/f & j \\ c & g & 1/k \end{bmatrix}$

E. $\begin{bmatrix} 1/a & 1/b & 1/c \\ 1/e & 1/f & 1/g \\ 1/i & 1/j & 1/k \end{bmatrix}$

F. $\begin{bmatrix} 1/a & 1/e & 1/i \\ 1/b & 1/f & 1/j \\ 1/c & 1/g & 1/k \end{bmatrix}$

G. None of the other options is the correct answer

(23) **[3 marks]** Given the above **view-transformation** matrix ***V***, which of the following is the matrix that can be used to transform a **3D vector** from **view space** to **world space**?

A. $\begin{bmatrix} a & b & c \\ e & f & g \\ i & j & k \end{bmatrix}$

B. $\begin{bmatrix} a & e & i \\ b & f & j \\ c & g & k \end{bmatrix}$

C. $\begin{bmatrix} 1/a & b & c \\ e & 1/f & g \\ i & j & 1/k \end{bmatrix}$

D. $\begin{bmatrix} 1/a & e & i \\ b & 1/f & j \\ c & g & 1/k \end{bmatrix}$

E. $\begin{bmatrix} 1/a & 1/b & 1/c \\ 1/e & 1/f & 1/g \\ 1/i & 1/j & 1/k \end{bmatrix}$

F. $\begin{bmatrix} 1/a & 1/e & 1/i \\ 1/b & 1/f & 1/j \\ 1/c & 1/g & 1/k \end{bmatrix}$

G. None of the other options is the correct answer

Section G [12 marks]

(24) **[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 `0000` and `0100`, what is the number of **intersections** that need to be computed?

- **A.** 0
- **B.** 1
- **C.** 2
- **D.** 3
- **E.** 4
- **F.** More than 4

(25) **[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 `0000` and `0101`, what is the number of **intersections** that need to be computed?

- **A.** 0
- **B.** 1
- **C.** 2
- **D.** 3
- **E.** 4
- **F.** More than 4

(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 `1001` and `0101`, what is the number of **intersections** that need to be computed?

- **A.** 0
- **B.** 1
- **C.** 2
- **D.** 3
- **E.** 4
- **F.** More than 4

(27) **[3 marks]** A **line segment** from the pixel location (100, 600) to (300, 200) is **scan-converted**. Assuming the line segment 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 line segment? This number includes the two fragments at the two endpoints of the line segment.

A. 200
B. 201
C. 400
D. 401
E. 600
F. 601

(28) **[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, and T_8 only. The triangles are rendered in the order T_1, T_2, …, T_8. Suppose the **depth values** of the fragments of T_1, T_2, …, T_8 at the location of P are 0.5, 0.9, 0.6, 0.4, 0.2, 0.8, 0.1, 0.3, respectively, how many times will the z-value in the **z-buffer** for pixel P be modified *after* it was initialized?

A. 1
B. 2
C. 3
D. 4
E. 5
F. 6
G. 7

——— END OF QUESTIONS ———