

Name: \_\_\_\_\_ MARKING KEY \_\_\_\_\_

Student ID Number: \_\_\_\_\_

Signature: \_\_\_\_\_

Fire Alarm Code: \_\_\_\_\_

## CPSC 414 97-98(T2) Midterm Exam #2

Department of Computer Science  
University of British Columbia  
K. S. Booth

### Exam Instructions (Read Carefully)

1. Print your **Name** and **Student Identification Number** on **every** page of the exam in the space provided at the top of each page **immediately**.
2. Sign the first page of the exam with your **Signature** in the space provided on the upper left **immediately**.
3. Continue reading the instructions, but **do not open the exam booklet** until you are told to do so by a proctor.
4. Cheating is an academic offense. Your signature on the exam indicates that you **understand** and **agree to** the University's policies regarding cheating on exams.
5. Please read the **entire** exam before answering any of the questions.
6. There are three questions on this exam, each worth the indicated number of marks. **Answer as many questions as you can.**
7. Write **all** of your answers on these pages. If you need more space, there is blank space at the end of the exam. Be sure to indicate when a question is continued, **both** on the page for that question and on the continuation page.
8. Interpret the exam questions as written. **No questions** will be answered by the proctors during the exam period.
9. The exam is **closed book**. You may not use any aids.
10. You have 45 minutes in which to work. **Budget your time wisely.**
11. In the event of a **fire alarm** during the exam, enter the four-character code provided by the proctor(s) in the space on the upper right, then gather your belongings and exit the room, handing your exam to a proctor as you exit.
12. No one will be permitted to leave the exam room during the **last ten minutes** of the exam.

No.	Avg	Max	Marker
1(a)		20	Dave
1(b)		20	Dave
2(a)		4	Miranda
2(b)		4	Miranda
2(c)		4	Miranda
2(d)		4	Miranda
2(e)		4	Miranda
3(a-f)		12	Dave
3(g-m)		14	Davor
3(n-t)		14	Davor
<b>Total</b>		<b>100</b>	

**Question #1 (40 marks – 2 marks each)**

This question tests your knowledge of geometric transformations, coordinate systems, hierarchical modeling, and some of the concepts used in the Alias|wavefront animation software. For each of the following, check the appropriate box to indicate whether the statement is generally **True** or generally **False**. Wrong answers count negative, so don't guess.

True	False	Statement (+2 for correct answers, –2 for incorrect)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	A <i>dolly</i> operation in Alias wavefront changes the size of a selected object.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Selection of a <i>pivot</i> in Alias wavefront affects how a subsequent scaling or rotation operation will be interpreted.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Unlike a general <i>DAG</i> in which each node can have many pointers to other nodes, in a <i>threaded DAG</i> each node has at most two pointers to other nodes.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	In a <i>threaded DAG</i> every child of a node has a pointer to its parent in the hierarchical scene description.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	In a <i>threaded DAG</i> every child of a node is pointed to by its parent in the hierarchical scene description.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	In the <i>synthetic camera</i> analogy, the near and far clipping planes could be thought of as similar to <i>depth-of-field</i> because they cause objects that are “too close” or “too far away” to be invisible (in a real camera they would be “out of focus”); however, a correct depth-of-field calculation is much more complicated than simply setting near and far clipping planes.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	In SPHIGS and OpenGL every <i>viewport</i> must have a corresponding <i>window</i> .
<input type="checkbox"/>	<input checked="" type="checkbox"/>	In SPHIGS and OpenGL the <i>colour</i> attribute is inherited from the parent and changes are only transmitted to descendants lower in the hierarchical scene description.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	In SPHIGS and OpenGL the <i>current transformation matrix</i> (or its equivalent) is inherited from the parent and changes are only transmitted to descendants lower in the hierarchical scene description.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	It is always possible to determine whether a transformation is a <i>modeling transformation</i> or a <i>viewing transformation</i> by examining its matrix representation.

True	False	Statement (+2 for correct answers, -2 for incorrect)
<input checked="" type="checkbox"/>	<input type="checkbox"/>	In computer graphics, most of the time algorithms assume that polygons are <i>planar</i> and thus they do not bother to check if this is actually the case.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	There are only five <i>degrees of freedom</i> when specifying the position and orientation of a synthetic camera, just like for a real camera.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Given the coordinates for a point, it is always possible to tell the <i>dimensionality</i> of the point (i.e., simply counting the number of coordinates will tell if the point is 2-D or 3D).
<input checked="" type="checkbox"/>	<input type="checkbox"/>	In a <i>right-handed</i> coordinate system the positive z-axis comes “out” of the screen if the positive x-axis points to the right and the positive y-axis points up.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	In a <i>left-handed</i> coordinate system, positive rotations are counter-clockwise when we are looking from the positive axis toward the origin.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	For 2-D transformations, because we do not have to deal with perspective, we can represent any transformation that is used in computer graphics by storing just <i>six</i> of the <i>nine</i> elements of the homogeneous matrix that represents the transformation.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	In 2-D viewing, a window and corresponding viewport must always have the same <i>aspect ratio</i> , but they need not be the same size.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Every <i>affine</i> transformation is a <i>rigid body</i> transformation.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	A <i>perspective matrix</i> can never be used as a modeling transformation, only as a viewing transformation.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	All rotation transformations in 3-D can be represented by <i>special orthogonal</i> matrices.

**Question #2 (20 marks – 4 marks each)**

This question tests your knowledge of 2-D and 3-D transformations and their various representations using vectors and matrices.

- (a) Write the *symbolic*  $4 \times 4$  matrix that represents a rotation about the  $y$ -axis by an angle of 120 degrees in the *clockwise* direction (i.e., so that a point on the positive  $x$ -axis moves one-third of the way past the positive  $z$ -axis). Be sure that you observe all of the conventions regarding the direction of rotation and the order in which matrices and vectors are multiplied together. All of the entries in your matrix should be zeros, ones, or sines and cosines of 120 degrees.

$$\begin{bmatrix} \cos(120) & 0 & -\sin(120) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(120) & 0 & \cos(120) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

OR

$$\begin{bmatrix} \cos(-120) & 0 & \sin(-120) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-120) & 0 & \cos(-120) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- (b) Re-write your answer to (a) with numeric entries, not symbolic (i.e., do not have unevaluated sine or cosine functions). You may leave numeric square roots in the entries. (**Note:** A right triangle with a 60 degree angle and a hypotenuse of length one has one side of length one half.)

$$\begin{bmatrix} -\frac{1}{2} & 0 & -\frac{\sqrt{3}}{2} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{\sqrt{3}}{2} & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- (c) The matrix **Q** below represents a modeling transformation that might be used within computer graphics. Describe *in words* what the transformation does by explaining how a *unit cube* centred on the origin would be transformed by the matrix **Q**.

$$\mathbf{Q} = \begin{bmatrix} 35 & 0 & 0 & 14 \\ 0 & 1 & 0 & -28 \\ 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 7 \end{bmatrix}$$

The cube is scaled to become a rectangular parallelepiped with dimensions scaled by 5 in  $x$  (so it is fatter), scaled by  $1/7$  in  $y$  (so it is skinnier) and scaled by 1 in  $z$  (so it stays the same); it is then moved so that the centre is at the point  $(2, -4, 0)$ , which is two units to the right and four units lower but at the same distance away in  $z$ .

- (d) What are the non-homogeneous representations for the two homogeneous  $(x, y, w)$  points whose coordinates are  $[7, -5, 2]^T$  and  $[-7, 5, 0]^T$ , respectively? Be sure to explain any anomalous situations if they exist.

The first point is  $[3.5, -2.5]^T$  (or  $[\frac{7}{2}, -\frac{5}{2}]^T$ ) in non-homogeneous coordinates but the second point has no representation in non-homogeneous coordinates because it is a point at infinity.

- (e) We know that in general matrices do not commute. Fill in the following table with one of “A”, “S” or “N” to indicate whether the corresponding pairs of matrices

<i>always</i>	commute =	<b>A</b>
<i>sometimes</i>	commute =	<b>S</b>
<i>never</i>	commute =	<b>N</b>

For purposes of this question, a “non-trivial” rotation is one that is not an integer multiple of  $\frac{\pi}{2}$ .

Types of Matrices	Commutativity
two non-uniform scalings	<b>A</b>
two non-trivial homogeneous 2-D rotations	<b>A</b>
two non-homogeneous 3-D rotations	<b>S</b>
a translation and a uniform scaling	<b>S</b>
a non-trivial 3-D rotation and a non-uniform scaling	<b>S or N</b>

Each wrong answer above is -1 mark (but you can't go negative).

**Question #3 (40 marks – 2 marks each)**

Explain the following terms as they have been used so far in our course on computer graphics. If the term is an *acronym*, explain for what it is an acronym *and* its meaning, usage, or how it functions (i.e., do not just provide the full name, but also explain what the acronym identifies).

- (a) center of projection

The point through which all ``lines of projection`` pass when a perspective transformation is applied.

- (b) concatenate

The multiplication or composition of two matrices or transformations.

- (c) DAG

A ``Directed Acyclic Graph`` is the most common form of hierarchical scene description used in computer graphics.

- (d) Frenet frame

For a 3-D curve, the Frenet frame is a coordinate system at a point on the curve whose principal axes are aligned with the tangent to the curve, the normal and the binormal for the curve.

- (e) glBegin()

An OpenGL function that specifies the beginning of a group of geometric primitives that follow the call to glBegin().

- (f) glEnd()

An OpenGL function that specifies the end of the group geometric primitives begun with a call to glBegin().

(g) homogeneous coordinates

In either 2-D or 3-D, the ``trick'' of adding an extra coordinate  $w$ , to that  $(x, y, z)$  becomes  $(wx, wy, wz, w)$  for  $w \neq 0$ .

(h) instance

A ``copy'' of a master object, often after one or more modeling transformations have been applied.

(i) inverse (of a matrix)

If  $A$  is a square matrix,  $A^{-1}$  is the unique matrix such that  $AA^{-1} = A^{-1}A = I$  where  $I$  is the identity matrix.

(j) master

The ``original'' geometric description of an object, usually defined in a convenient coordinate system that is independent of world coordinates. Multiple copies or ``instances'' of this can then be used to build scenes.

(k) pick ID

An integer (in OpenGL) or other ``tag'' that identifies an object or a collection of objects during a picking operation. In OpenGL, multiple names (integers) are returned that collectively serve as the pickID.

(l) point at infinity (in 3-D)

In homogeneous coordinates, when the homogeneous coordinate  $w=0$  a point has no finite representation, but instead is ``at infinity'' in the direction specified by its other coordinates.

(m) shear matrix (along the  $x$ -axis)

$$M_{shear} = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

so that  $x' = x + ay$  and  $y' = y$ , where  $a$  is any scalar constant.

(n) translation

A rigid body displacement or movement that preserves orientation.

(o) transpose of  $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$

For 2 marks:  $\begin{bmatrix} a & c \\ b & d \end{bmatrix}$  .

For 1 mark:  $\begin{bmatrix} a & b \\ c & d \end{bmatrix}^T$  .

(p) uniform scaling

A scaling transformation in which the  $x$ ,  $y$  and  $z$  scale factors are the same.

(q) unit tangent vector

The normalized first derivative of a vector-valued function  $f(t)$ , denoted by  $\hat{T}(t) = \frac{f'(t)}{|f'(t)|}$ . If  $f'(t)$  is zero, the unit tangent vector is undefined.

(r) viewport

A (usually) rectangular region on the screen in which a window is displayed.

(s) void Walk( node \*p );

The generic traversal function for a (threaded) DAG or hierarchical scene description.

(t) What is “for all  $a, b \in F$  and  $u, v \in V$ ,  $T(au + bv) = aT(u) + bT(v)$ ” ?

The fundamental axiom for a linear transformation on a vector space  $V$  over a field  $F$ .



Name: \_\_\_\_\_ MARKING KEY \_\_\_\_\_

Student ID Number: \_\_\_\_\_

*(extra space to continue work)*