Exam Instructions (Read Carefully)

1. Sign the first page of the exam with your Signature in the space provided on the upper left immediately.
2. Continue reading the instructions, but do not open the exam booklet until you are told to do so by a proctor.
3. Print your Name and Student Identification Number on every page in the space provided at the top of each page before you start the exam.
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 four 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 proctor(s) during the exam period.
9. The exam is closed book. There are no aids permitted of any kind.
10. You have 70 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.

Question | Mean | Maximum |
--- | --- | --- |
1(a) | 8.5 | 12 |
1(b) | 2.6 | 8 |
2 | 14.9 | 20 |
3(a) | 2.5 | 4 |
3(b) | 3.1 | 4 |
3(c) | 3.1 | 4 |
3(d) | 5.9 | 8 |
4(a)-(j) | 13.0 | 20 |
4(k)-(t) | 17.3 | 20 |
**Total** | **70.8** | **100**
Question #1 [20 marks total]
This question tests your knowledge of hierarchical display list structures as discussed in lecture.

Below is a DAG that will display multiple instances of the unit cube whose geometry is stored in the node "C" when the `Walk()` function is called to process the root node of the DAG. Refer to this diagram for both parts of this question.

The nodes with a "T" label are translations.

- "T0" is the translation that takes \((x, y, z)\) to \((x, y - 5, z)\).
- "T1" is the translation that takes \((x, y, z)\) to \((x, y, z)\).
- "T2" is the translation that takes \((x, y, z)\) to \((x, y + 5, z)\).
- "T3" is the translation that takes \((x, y, z)\) to \((x - 5, y, z)\).
- "T4" is the translation that takes \((x, y, z)\) to \((x, y, z)\).
- "T5" is the translation that takes \((x, y, z)\) to \((x + 5, y, z)\).

The "R" node is a counterclockwise rotation of 45 degrees about the x axis, and the "S" node is a scaling that expands everything by a factor of five along the x axis, contracts everything by a factor of two along the y axis, and leaves things unchanged along the z axis.

\[
\begin{array}{c}
\text{T0} \\
\text{T1} \\
\text{T2} \\
\text{T3} \\
\text{T4} \\
\text{T5} \\
\text{S} \\
\text{R} \\
\text{C}
\end{array}
\]

**NOTE:** This question was not exactly what was intended. The "S" and the "R" nodes were interchanged, which means that the rotation happens before the scaling. This means that the unit cubes (when scaling is applied) turn into parallelepipeds, not rectangular prisms (the difference is that opposite sides are still parallel to each other, but adjacent sides are not at right angles to each other -- this is a consequence of the combination of rotation and scaling being a general affine transformation, which preserves parallelism, but not angles).

A second problem is that the question refers to the "center of the screen". This assumes that \((0,0)\) is the \((x,y)\) center of the screen (which is the case for normal clipping). Some students may have a trouble with this because of the ambiguity about the center of the screen.

For this reason, the marking will be a bit lenient.
When a cube is drawn after a non-uniform scaling has been applied, it will in general be a rectangular prism, not a cube. If we assume that the function `Walk()` is invoked with `T0` as its argument (the root of the DAG), there will be six prisms (possibly scaled and rotated cubes) draw at various places on the screen. One of them will be at the center of the screen.

(a) [12 marks] For each of the six prisms, describe its position on the screen, its orientation, and its shape.

1. Located at \((-5, -5, 0)\) and rotated 45° about \(x\)-axis, this is the only one of the six that is not a cube; it is stretched by a factor of five along \(x\)-axis (so it is fatter), and compressed by one-half along the \(y\)-axis (so it is squatter -- a “diamond” shape that is half as tall as it is wide).

2. Located at \((0, -5, 0)\) and rotated 45° about \(x\)-axis, but still a unit cube.

3. Located at \((+5, -5, 0)\) with no rotation and still a unit cube.

4. Located at the center of the screen \((0, 0, 0)\) and rotated 45° about \(x\)-axis, but still a unit cube.

5. Located at \((+5, 0, 0)\) with no rotation and still a unit cube.

6. Located at \((+5, +5, 0)\) with no rotation and still a unit cube.

(b) [8 marks] Assume that the viewing transformation is just the identity matrix with no perspective. Show the contents of the matrix stack when the prism at the center of the screen is being drawn as the data structure is walked according to the `Walk()` algorithm described in lecture. The stack will contain five \(4 \times 4\) matrices with the current transformation on the top of the stack – you should show all of the symbolic expressions (various products of the matrices stored in the DAG) for each of the five matrices in the stack.

1. \(I \cdot T_1 \cdot T_4 \cdot R = T_1 \cdot T_4 \cdot R\)  
   **Note:** Order definitely counts!

2. \(I \cdot T_1 \cdot T_4 \cdot R = T_1 \cdot T_4 \cdot R\)

3. \(I \cdot T_1 \cdot T_4 = T_1 \cdot T_4\)

4. \(I \cdot T_1 = T_1\)

5. bottom: \(I\) (the identify matrix)

There is only one instance of the cube for which scaling is applied, so this is not a big deal, but it will affect your answer for the very first instance.

In general, the order in which you list the six instances of the cube is not important. What counts is that you get at least one object in each of the six positions, that one of them is scaled and rotated, two are only rotated, and three have neither rotation nor scaling. You get full marks if you have this and all in the correct positions. Any indication that the cube is deformed counts as "scaling”.

The correct cube for Part (b) is the one whose root-to-leaf path has the labels `T1-T4-R-C`. This means that there will be four (recursive) calls to `Walk()`, resulting in four matrices "on the stack" in addition to the "top of stack" current transformation matrix (a total of five).
Question #2 [20 marks total]
This question tests your knowledge of the properties of polygonal objects and the various types of coordinate that can be used to describe the vertices of a polygon.

In each of the five parts that follow, a polygon is described by its vertices, which are listed according to the usual convention of counter-clockwise order (starting with the first vertex in the list) as seen from the front side of the polygon.

Fill in the following table with a “Yes” or a “No” in each entry for each of the five polygons (a)-(e).

<table>
<thead>
<tr>
<th></th>
<th>Polygon (a)</th>
<th>Polygon (b)</th>
<th>Polygon (c)</th>
<th>Polygon (d)</th>
<th>Polygon (e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>planar</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>convex</td>
<td>No</td>
<td>Yes No Y</td>
<td>es No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>simple</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>front-facing</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

You may assume that the eye is at the point (0,0,1000) and that all of the vertices for 2-D polygons have $z = 0$.

Fill in “Yes” if the polygon has the property and “No” if it does not. If you think that the property is not well-defined for some polygon, you should answer “No”.

(a) A polygon whose 2-D non-homogeneous coordinates are: (2,2), (8,2), (4,4), and (2,8).

(b) A polygon whose 2-D homogeneous coordinates are: (2,2,2), (8,2,8), (4,4,16), and (4,8,8).

(c) A polygon whose 2-D homogeneous coordinates are: (2,2,2), (4,4,16), (8,2,8), and (2,8,8).

(d) A polygon whose 3-D non-homogeneous coordinates are: (2,0,0), (0,2,0), and (0,0,2).

(e) A polygon whose 3-D non-homogeneous coordinates are: (0,0,0), (2,0,0), (0,2,0), and (0,0,2).

The "No" answers are because: (a) is a concave "V"; (b) is seen counterclockwise and thus is back-facing; (c) is a "bow tie" and thus not simple and hence not convex; (d) is the origin plus three points, one along each of the principal axes, and thus it is not planar and consequently none of the other properties make sense.
Question #3 [20 marks total]
This question tests your knowledge of 2-D and 3-D transformations and their various representations using vectors and matrices.

(a) [4 marks] Write the $2 \times 2$ matrix that represents a rotation in the plane by an angle of 60 degrees in the clockwise direction (i.e., so that a point on the positive $y$-axis moves two-thirds of the way toward the positive $x$-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 numeric, not symbolic (i.e., do not have unevaluated sin or cos 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.

\[
\begin{bmatrix}
\cos(-60^\circ) & -\sin(-60^\circ) \\
\sin(-60^\circ) & \cos(-60^\circ)
\end{bmatrix} = \begin{bmatrix}
\cos(60^\circ) & \cos(60^\circ) \\
-\sin(60^\circ) & \cos(60^\circ)
\end{bmatrix} = \begin{bmatrix}
\frac{1}{2} & \frac{\sqrt{3}}{2} \\
-\frac{\sqrt{3}}{2} & \frac{1}{2}
\end{bmatrix} = \begin{bmatrix}
0.5 & 0.866... \\
-0.866... & 0.5
\end{bmatrix}
\]

The first two matrices are not good answers: one mark for the first matrix, two marks for the second, but they have to be exact to get any marks at all. Both the third and fourth matrices get the full four marks if they are correct, or one mark off for each incorrect entry (numeric signs must be correct or marks off).

(b) [4 marks] The matrix $Q$ below represents a 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 $Q$.

Note: The lower-righthand entry of the matrix is not a one. Be sure that you take this into account in your answer.

\[
Q = \begin{bmatrix}
4 & 0 & 0 & 8 \\
0 & 1/6 & 0 & -6 \\
0 & 0 & -10 & 0 \\
0 & 0 & 0 & 2
\end{bmatrix}
\]

The 2 in the lower right entry means that upon division by $w$ everything gets halved. This means $Q$ is equivalent to a matrix all of whose entries are half as big as those in $Q$. So the result of applying $Q$ is: (a) the unit cube becomes a rectangular prism; (b) its length along $x$ is doubled, its height along $y$ is reduced by a factor of twelve, and its depth along $z$ is increased by a factor of five; (c) the prism is moved so its center is located at the points $(4, -3, 0)$; and (d) the prism is turned "inside out" because the scaling about $z$ is negative.
(c) [4 marks] What are the 2-D non-homogeneous representations for the 2-D homogeneous \((x, y, w)\) points whose coordinates are the following:

1. \([13, -8, 2]^T\) converts to non-homogeneous \([6.5, -4]^T\)

2. \([-13.8, 0]^T\) does not convert to anything because it is a "point at infinity" \((w=0)\) and thus it is a direction, not a point, and has not non-homogeneous representation as a point although it does have a vector representation as \([-13.8]^T\) or any scalar multiple of this.

If there are problems converting to the non-homogeneous representation, explain why this is the case and how to interpret the result.

(d) [8 marks] We know that in general matrices do not commute. Fill in the second column of following table with exactly one of “A”, “S” or “N” to indicate whether the corresponding pairs of matrices always, sometimes or never commute.

<table>
<thead>
<tr>
<th>Types of Matrices</th>
<th>Commutativity</th>
</tr>
</thead>
<tbody>
<tr>
<td>two non-uniform scalings</td>
<td>A</td>
</tr>
<tr>
<td>a translation and a non-uniform scaling</td>
<td>S</td>
</tr>
<tr>
<td>a uniform scaling and a translation</td>
<td>S</td>
</tr>
<tr>
<td>a 3-D rotation and a uniform scaling</td>
<td>A</td>
</tr>
<tr>
<td>two homogeneous 2-D rotations</td>
<td>A</td>
</tr>
<tr>
<td>two non-homogeneous 2-D rotations</td>
<td>A</td>
</tr>
<tr>
<td>two homogeneous 3-D rotations</td>
<td>S</td>
</tr>
<tr>
<td>two non-homogeneous 3-D rotations</td>
<td>S</td>
</tr>
</tbody>
</table>

There are No answers of N because every type of matrices listed includes the identity matrix and these always commute with any other matrix of that type. So the only possible answers are A if the matrices always commute, or S if the matrices sometimes commute.

For each of the S answers there is an easy example of a pair of matrices that do not commute. For the A answers, the best "proof" is to multiply out the general symbolic forms (as suggested in the course newsgroup) and confirm that the products in either order are the same regardless of the actual values of the symbolic entries.
Question #4 [40 marks – 2 marks each]
This question tests your general knowledge of the concepts and terminology introduced in the course.

The following terms are possible answers for the questions on subsequent pages. Use the number corresponding to a term below as an answer if you think it is the best match for one of the concepts or terms on subsequent pages.

(1) DAG
(2) Euclidean space
(3) fluorescence
(4) forward dynamics
(5) forward kinematics
(6) half-life
(7) homogeneous coordinates
(8) inverse dynamics
(9) inverse kinematics
(10) inverse matrix
(11) joint angle
(12) key-frame animation
(13) left-handed coordinate system
(14) linear interpolation
(15) linear transformation
(16) orthographic
(17) perspective
(18) phosphorescence
(19) physically-based animation
(20) point at infinity
(21) quaternion
(22) refresh rate
(23) right-handed coordinate system
(24) rotation matrix
(25) translation matrix
(26) transpose matrix
(27) update rate
(28) w-coordinate
(29) window edge coordinate
For each definition or phrase below and on subsequent pages, write the number of the term listed on the previous page that best matches the definition or phrase.

[2 marks each]

_ 3 _ (a) Light emitted when molecules (such as phosphors) decay from excited states almost instantaneously after excitation.

_ 18 _ (b) Light emitted when molecules (such as phosphors) decay from excited states some time after excitation.

_ 12 _ (c) The specification of motion in an animation system by providing the location and orientation of all of the objects in the scene at specific times in the animation sequence and then letting the computer algorithmically generate all of the “in-between” frames.

_ 9 _ (d) A procedure to determine the joint angles required for a desired end effector position and orientation.

_ 14 _ (e) An “in-betweening” technique in which a first-order formula \( P(t) = P_0 + t \cdot (P_1 - P_0) \) is used to generate intermediate positions.

_ 5 _ (f) A procedure to determine the position and orientation of an end effector, given the joint angles.

_ 19 _ (g) Any procedure that uses physics as a constraint for motion control.

_ 4 _ (h) A procedure that determines the accelerations on each part of an object given the internal and external forces and torques that are applied to the object in an animation.

_ 21 _ (i) An algebraic entity something like a vector, except with four components instead of three, that can be used to represent rotations in three-dimensional space. (Discovered by Sir William Rowen Hamilton in 1843 while walking with his wife one evening along the Royal Canal in Dublin.)

_ 24 _ (j) A commonly used transformation that always has a special orthogonal matrix representing it.
[2 marks each]

_ 22 _ (k)  The number of times per second that an image is displayed on a screen to maintain the illusion of a continuous image, known as “persistence of vision”. Usually 60-72 Hz is required, but sometimes only 40 Hz is sufficient for the “critical fusion frequency”.

_ 27 _ (l)  The number of times per second that the image being displayed on a screen is modified to produce the illusion of continuous, smooth motion. Usually 30-72 Hz is required, but sometimes only 10-15 Hz is sufficient.

_ 7 _ (m)  In either 2-D or 3-D, the “trick” of adding an extra coordinate \( w \), so that \((x, y, z, w)\) becomes \((wx, wy, wz, w)\) for \( w \neq 0 \).

_ 13 _ (n)  The positive \( x \)-axis points to the right, the positive \( y \)-axis points upward, and the positive \( z \)-axis points away from the viewer when the viewer looks at the origin.

_ 20 _ (o)  In homogeneous coordinates, when the homogeneous coordinate \( w = 0 \), a point has no finite representation, but instead is considered to be a vector in the direction specified by its other coordinates.

_ 16 _ (p)  A parallel projection in which all projection lines are normal to the projection plane.

_ 28 _ (q)  An extra non-zero scaling factor that is added as an \( n^{th} \) component of all vectors and then the other components are multiplied by this new component.

_ 26 _ (r)  Given a matrix with entries \((a_{ij})\), this is a related matrix whose entries are \((a_{ji})\).

_ 15 _ (s)  Any mapping that satisfies the property:

\[
T(au + bv) = aT(u) + bT(v)
\]

where \( F \) is a field and \( V \) is a vector space over that field.

_ 1 _ (t)  A data structure often used to describe a hierarchical scene by having interior nodes represent transformations that apply to descendent nodes (nodes that can be reached by a directed path from the ancestor node); it differs from a tree because it permits substructures (objects) to be included multiple times with different transformations applied to each instance.