A. Vectors (4 points)

Given the following two vectors:

\[
\begin{align*}
\vec{a} &= \begin{bmatrix} 3 \\ 4 \\ 0 \end{bmatrix} \\
\vec{b} &= \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}
\end{align*}
\]

Compute the following:

1. The magnitude of \(\vec{a}\) 5
2. The magnitude of \(\vec{b}\) 1
3. The dot product \(\vec{a} \cdot \vec{b}\) 4
4. The cosine of the angle between \(\vec{a}\) and \(\vec{b}\) \(4/5\) AKA .8

B. Normal (1 point)

You are looking down at a flat quadrilateral, defined by four points that all lie in a plane:

5. Write an expression for the normal to this quadrilateral. The normal should be pointing towards you (out of the paper).
   \[(b-a) \times (d-a) / |(b-a) \times (d-a)| \text{ and similar}\]
C. Point and vector expressions (6 points)

For each of the following expressions involving points \( \mathbf{a} \) and \( \mathbf{b} \) and vectors \( \mathbf{u} \) and \( \mathbf{v} \), circle “Point” if the result is a point, “Vector” if the result is a vector, or “Invalid” if the expression is not legal (AKA not affine invariant). Hint: consider what happens to the \( w \) component in homogeneous coordinates.

6. \( \mathbf{a} + \mathbf{b} \) Point Vector Invalid
7. \( \mathbf{a} - \mathbf{b} \) Point Vector Invalid
8. \( \mathbf{u} - \mathbf{v} \) Point Vector Invalid
9. \( \mathbf{a} + 2\mathbf{u} \) Point Vector Invalid
10. \( \frac{1}{3}\mathbf{a} + \frac{2}{3}\mathbf{b} \) Point Vector Invalid
11. \( \frac{1}{3}\mathbf{a} + \frac{2}{3}\mathbf{v} \) Point Vector Invalid

D. Transformation matrix (2 points)

Consider a homogeneous affine transformation matrix \( \mathbf{M} \), constructed from columns \( \mathbf{\bar{a}}, \mathbf{\bar{b}}, \mathbf{\bar{c}} \) and \( \mathbf{\bar{d}} \):

\[
\begin{bmatrix}
  a_x & b_x & c_x & d_x \\
  a_y & b_y & c_y & d_y \\
  a_z & b_z & c_z & d_z \\
  0 & 0 & 0 & 1
\end{bmatrix}
\]

12. Write the result of transforming \[
\begin{bmatrix}
  0 \\
  1 \\
  0
\end{bmatrix}
\]
by \( \mathbf{M} \), in terms of \( \mathbf{\bar{a}}, \mathbf{\bar{b}}, \mathbf{\bar{c}} \) and \( \mathbf{\bar{d}} \). \( \mathbf{b+d} \)

13. Suppose this matrix represents the local-to-world transform of an object that is not changing position, but is spinning about its local z-axis. Which column or columns ( \( \mathbf{\bar{a}}, \mathbf{\bar{b}}, \mathbf{\bar{c}} \) or \( \mathbf{\bar{d}} \) ) of the matrix will not change as the object spins? 

\( c \) and \( d \)
E. Scene Modeling (4 points)

14. Write the local-to-world transform that places the boat (left) in world coordinates in the scene (right), expressed as a composition of 2D translation $T(x, y)$ and rotation $R(\text{angle})$. Give numbers for the parameters of the transformations you use.

$$T(3, 4) \ R(-45)$$

In this scene, a sailor and a palm tree are on the deck of the boat. The sailor’s bird is on a branch of the tree. Write the following using composition of the local transformation matrices $T_{\text{boat}}, R_{\text{boat}}, T_{\text{deck}}, T_{\text{sailor}}, T_{\text{palm}},$ and $T_{\text{bird}}$ and their inverses as needed:

15. The object-to-world transform for the bird $T_{\text{boat}} \ R_{\text{boat}} \ T_{\text{deck}} \ T_{\text{palm}} \ T_{\text{bird}}$

16. The object-to-world transform for the sailor $T_{\text{boat}} \ R_{\text{boat}} \ T_{\text{deck}} \ T_{\text{sailor}}$

17. The transform expressing the bird’s frame in the sailor’s coordinates

$$T_{\text{sailor}}^{-1} \ T_{\text{palm}} \ T_{\text{bird}}$$
F. Concepts (8 points)
   a. Camera Space
   b. Image Space
   c. Inner Space
   d. Normalized View Space
   e. Object Space
   f. Outer Space
   g. Projective Space
   h. World Space

18. Graphics pipeline: Select the correct spaces from above and list them in the order that we expect each vertex to go through in the traditional transformation process:

   Fill in letters here: 1.  e  2.  h  3.  a  4.  d  5.  b

Match the descriptions to the appropriate concept.
   a. A simpler object to make culling tests quicker
   b. Aim at an object of interest
   c. Draw or don’t draw triangles based on the order of their vertices
   d. Makes things farther away appear smaller
   e. Only draw objects that are in view
   f. Preserves parallel lines

19.  c  Backface culling
20.  a  Bounding volume
21.  e  Frustum culling
22.  b  Look-at transform
23.  f  Orthographic transform
24.  d  Perspective transform

25. The following pseudocode operations are the steps involved in recursive traversal of a scene hierarchy. Arrange them in proper order for a `drawNode()` method.

   a. Apply local transform
   b. Draw geometry (if any)
   c. Recursively draw children (if any)
   d. Pop Matrix
   e. Push Matrix

   ```
   drawNode(node) {
       _e_
       _a_
       _b_ or c
       _c_ or b
       _d_
   }
   ```