1. Coordinate Frames

(a) (3 points) Express point \( P \) and vector \( V \) in each of the three coordinate frames.

\[
\begin{align*}
P_1 & : (-2, -4) & V_1 & : (-2, 1) & P_2 & : \left( \frac{2}{3}, \frac{2}{3} \right) \\
P_2 & : (2, 3) & V_2 & : (-1, -2) & V_3 & : (-1, 1)
\end{align*}
\]

\[
F_3: \text{ move right 1 unit: } P_3: 4 \left( \frac{2i - i}{3} \right) - 1 \left( \frac{2j - i}{3} \right) = \frac{8}{3}i + \frac{4}{3}j - \frac{5}{3}j + \frac{1}{3}i
\]

(b) (2 points) Find the \( 3 \times 3 \) homogeneous transformation matrix which takes a point from \( F_3 \) and expresses it in terms of \( F_1 \). I.e., determine \( M \), where \( P_1 = MP_3 \).

See above.
Express \( i_3, j_3, o_3 \) in terms of Frame \( F_0 \).

(c) (2 points) Give the OpenGL transformations, i.e., the sequence of translates, scales, and rotations, that implements the transformation \( M \), where \( P_2 = MP_1 \). Assume that \( z \) remains unchanged.

Think of \( F_1 \) as being the object frame and \( F_2 \) as being the world frame.

\[
\begin{bmatrix}
glLoadIdentity () \quad & \text{(optional for this question)} \\
glTranslate (-2, 1, 0) \\
glRotate (90^\circ, 0, 0, 1) \quad \text{align x-axis, i.e., } i \text{ downwards}
\end{bmatrix}
\]

\[
glScale (1, -1, 1) \quad \text{flip y}
\]
2. Short Answer

(a) (1 point) Express the point \((2,3,4,2)\), given in homogenous coordinates, in cartesian coordinates.

\[
\begin{pmatrix} x, y, z, w \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} x/w, y/w, z/w \end{pmatrix} = \left( \frac{3}{2}, \frac{3}{2}, \frac{4}{2} \right) = (1, 1.5, 2)
\]

(b) (1 point) How many numbers are needed to specify an arbitrary 3D affine transformation?

Rotations, scales, shears, and translations each require numbers, so 12 numbers.

(c) (1 point) What is the inverse of the following matrix?

\[
\begin{bmatrix} -1 & 0 & 0 & -3 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\]

Inverse:

\[
\begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\]

(d) (1 point) How does one “zoom in” using a pinhole camera? I.e., What should one do or modify to create an image with a more detailed, enlarged view of the world?

Move the image plane back.

(e) (1 point) What is the disadvantage of using a large aperture with a real camera?

This limits the depth of field, i.e., the range for which objects are in focus.

(f) (1 point) True or False: Humans have an upside-down image of the world on their retina.

False

(g) (1 point) Describe how to compute the angle between two arbitrary 3D vectors, \(a\) and \(b\).

\[
\theta = \cos^{-1} \left( \frac{a \cdot b}{|a||b|} \right) \iff a \cdot b = |a||b| \cos \theta
\]

(h) (1 point) True or False: It is possible to take a single photograph that simultaneously has various different objects shown in one-point, two-point, and three-point perspective.

Multiple cubes:

(i) (3 points) Give implicit, explicit, and parametric equations for the line passing through points \(P_a(x_a, y_a)\) and \(P_b(x_b, y_b)\).

\[
\begin{align*}
&\text{parametric:} \\
P(t) &= P_a + t(P_b - P_a) \\
\begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} x_a \\ y_a \end{bmatrix} + t \begin{bmatrix} x_b - x_a \\ y_b - y_a \end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
&\text{implicit:} \\
\text{slope} &= \frac{y_b - y_a}{x_b - x_a} \\
\Rightarrow \quad D &= (y_b - y_a)(x - x_a) - (y_b - y_a) \cdot (x - x_a)
\end{align*}
\]

\[
\begin{align*}
&\text{explicit:} \\
y &= y_a + \frac{y_b - y_a}{x_b - x_a} \cdot (x - x_a)
\end{align*}
\]

Other solutions possible!
(j) (2 points) Consider an object that is to be drawn using the ModelView matrix $M$. (i) How would the drawn scene change if $M$ were left-multiplied by a translation matrix, $T = \text{Trans}(1,0,0)$, i.e., $M' = TM$. (ii) How would it change if $M$ were right-multiplied, i.e., $M' = MT$?

(i) Scene moves to the right one unit in viewing coordinates

(ii) Scene moves to the right one unit in local object coordinates.

3. Scene Graphs

![Scene Graph Diagram]

(a) (2 points) Give the transformation that would take points from frame $E$ and re-express them in frame $D$.

$$P_D = M_D^{-1} M_A^{-1} M_B M_E P_E$$

(b) (2 points) The code given below is used to prepare to draw object $C$ in the above scene graph. Compute the value of $M_C$.

```c
glRotatef(90,0,0,1);
glTranslatef(2,2,2);
glScalef(1,0,0);
```

$$\begin{bmatrix}
0 & -1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}$$

$$\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}$$

$$\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}$$

$$= \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}$$
4. (6 points) Give the OpenGL code needed to draw the following scene of a simple animated arm. The three links should be drawn by making calls to `drawCube()`. Assume that the current transformation matrix is initialized to the identity matrix. Your code should leave the transformation stack unchanged after its execution. Block A has dimensions \(2 \times 4\) and blocks B and C have dimensions \(1 \times 3\). Blocks B and C rotate about the points shown.

```c
drawCube() {
    glBegin(GL_LINE_LOOP);
    glVertex2f(-1,0);
    glVertex2f(1,0);
    glVertex2f(1,1);
    glVertex2f(-1,1);
    glEnd();
}
```

draw scaled cube (float \(a\), float \(b\))

```c
    glPushMatrix();
    glTranslatef(\(a\), \(b\), 0);
    draw Cube();
    glPopMatrix();
```

This is one possible solution of many.
5. Viewing and Projection Transformations

(a) (3 points) Answer the following questions with true or false. Each correct answer is worth +1 and each incorrect answer is worth -1, so guessing is not encouraged.

- The projection matrix is defined by the view frustum parameters. \[ \text{T} \]
- The view volume in normalized device coordinates is defined by: \[ 0 \leq x \leq 1, 0 \leq y \leq 1, 0 \leq z \leq 1. \]
- Multiplying each of left, right, top, bottom in a view frustum specification by 0.5 will result in a zoom effect, i.e., the objects at the center of the image will appear larger. \[ \text{T} \]

(b) (1 point) Does the following projection matrix implement a parallel projection or a perspective projection? Explain your reasoning.

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & -5/3 & -8/3 \\
0 & 0 & -1 & 0
\end{bmatrix}
\]

\[ \text{this alters the value of } h, \quad \text{i.e., } h' = -2 \]

(c) (1 point) Will the VCS point \( P(1, 3, -2) \) appear on screen with the above projection matrix? Show your work.

\[
\begin{bmatrix}
y' \\
h'
\end{bmatrix} =
\begin{bmatrix}
1 & 3 & 2/3 \\
0 & 2 & 2
\end{bmatrix}
\begin{bmatrix}
y \\
h
\end{bmatrix} =
\begin{bmatrix}
1/2 \\
1/3
\end{bmatrix}
\]

Therefore, \( h' < 1 \) outside view volume.

(d) (1 point) Will the VCS point \( P(1, 1, -5) \) appear on screen with the above projection matrix? Show your work.

\[
\begin{bmatrix}
x' \\
y' \\
h'
\end{bmatrix} =
\begin{bmatrix}
1 & 1 & 1/2 \\
0 & 1 & 1 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
h
\end{bmatrix} =
\begin{bmatrix}
1/2 \\
1/2 \\
1/2
\end{bmatrix}
\]

Therefore, \( h' > 1 \) outside view volume.
(e) (4 points) Suppose that you wish to have a user select an object, using a mouse, in a 1000 x 1000 image that has been rendered using a perspective projection. You are given the \((x, y)\) display coordinates of the point selected by the user. This point is representative of a line going into the scene, which can then be intersected with objects in world coordinates in order to determine which object was selected.

Give an expression for \(x_{\text{NDCS}}\) and \(y_{\text{NDCS}}\) as a function of \((x, y)\).

Determine the VCS coordinates of the point, assuming that it sits on the near plane and using the view frustum parameters of \(\text{near} = 1, \text{far} = 50, \text{left} = -1, \text{right} = 1, \text{bot} = -1, \text{top} = 1\).

Determine a second point in VCS that could be used to define a line equation in VCS, and give the resulting line equation.

Assuming that the viewing matrix, \(M_{\text{view}}\) is known, give the line equation in WCS.

\[
\begin{align*}
(x, y) & \rightarrow (x', y') \rightarrow (x_{\text{NDCS}}, y_{\text{NDCS}}) \rightarrow (x_{\text{VCS}}, y_{\text{VCS}}) \rightarrow (x_{\text{WCS}}, y_{\text{WCS}}) \\
(0,0,0,1) & \rightarrow (-\frac{x}{1000}, \frac{y}{1000}, \frac{z}{1000}, 1) \rightarrow (-x_{\text{VCS}}, y_{\text{VCS}}, -z_{\text{VCS}}, 1) \rightarrow (-x_{\text{WCS}}, y_{\text{WCS}}, -z_{\text{WCS}}, 1) \\
\end{align*}
\]

For the given values

\[
\begin{align*}
x_{\text{VCS}} & = x_{\text{NOCs}} + 1 \\
y_{\text{VCS}} & = y_{\text{NOCs}} \\
\text{because left, right, top, bot are defined on the near plane.} \\
\end{align*}
\]

A second point: \((0, 0, 0)_{\text{VCS}}\), i.e., the eye point.

\[
\begin{align*}
P_1 & = (0, 0, 0)_{\text{VCS}} \\
P_2 & = \left(\frac{x}{\text{S20}}, \frac{y}{\text{S20}}, -1, \text{near}\right)_{\text{VCS}} \\
P(\text{P1})_{\text{WCS}} & = M_{\text{view}}^{-1} \left( P_1 + \frac{1}{2} (P_2 - P_1) \right)
\end{align*}
\]