1. Transformations

(a) Suppose we want to animate a ball being juggled. The ball is moving in a circle. It’s \( z \) coordinate isn’t changing, but its \( x \) and \( y \) coordinates are changing so that it is moving in a circle centered at the origin. It begins at the bottom of the circle, one foot below the origin (at point \((0, -1, 0)\)). It is moving 360 degrees per second, and we are generating 30 frames per second. Describe a matrix that we can apply to update the position of the center of the ball from one frame to the next.

The ball is rotating around the \( z \) axis. We know this because the \( z \) coordinates don’t change, and the points on the axis of rotation don’t move when an object rotates. The ball moves 12 degrees per frame, so the appropriate rotation is:

\[
\begin{pmatrix}
\cos 12 & -\sin 12 & 0 & 0 \\
\sin 12 & \cos 12 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

Note that this doesn’t depend on how much time has passed or on where the ball started.
(b) Provide a matrix that we can apply if the juggler is also walking forward (in the z direction) at one foot per second. (If you can’t solve (a), for partial credit provide a matrix that describes the motion of the ball due to walking).

In addition to rotating, we want to move the ball forward at one foot per second (1/30 of a foot per frame). We can do this with:

\[
\begin{bmatrix}
\cos 12 & -\sin 12 & 0 & 0 \\
\sin 12 & \cos 12 & 0 & 0 \\
0 & 0 & 1 & \frac{1}{30} \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

This combines a rotation and translation. We can get this directly, by seeing that this changes x and y coordinates with a rotation, and z coordinates with a translation. Or we can combine two transformations:

\[
\begin{bmatrix}
\cos 12 & -\sin 12 & 0 & 0 \\
\sin 12 & \cos 12 & 0 & 0 \\
0 & 0 & 1 & \frac{1}{30} \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{30} \\
0 & 0 & 1 & 0
\end{bmatrix}
= 
\begin{bmatrix}
\cos 12 & -\sin 12 & 0 & 0 \\
\sin 12 & \cos 12 & 0 & 0 \\
0 & 0 & 1 & \frac{1}{30} \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & \frac{1}{30} \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

It doesn’t matter in which order we apply rotation and translation in this special case. When we translate in the direction of the axis of rotation, translation doesn’t change the rotation axis, so the rotation has the same effect before or after the translation.

2. Viewpoint

Suppose we want to show the world from the viewpoint of an observer at (2,3,0), looking up, in the y direction. She is lying down, so that up to her is in the z direction. Provide a matrix that will transform a point in the world into this viewer’s coordinates.

We need to translate by (-2,-3,0), so that the point (2,3,0) becomes the origin, (0,0,0).

We can do this with a matrix of the form:
We can change viewpoint with a second matrix. Our new z direction will be the old y direction, (0,1,0). To achieve this, we have:

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

This will mean that the new z coordinate will be the old y coordinate. Up to her, the new y direction, should be the old z coordinate, which we get with:

\[
\begin{pmatrix}
0 \\
0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

Note, it would have been ok if we’d used -1 for the element in row 2, column 3, since we didn’t specify whether up was the plus or minus z direction. Finally, to keep the rows of the rotation orthonormal, the last three values have to be either (1,0,0) or (-1,0,0). However, we have to use (-1,0,0) for the rotation part of the matrix to have a determinant of 1, instead of -1. Even without explicitly taking determinants, we can imagine rotating so that you face in the y direction, with the z direction up, and we’ll see that the x direction reverses. So we get:

\[
\begin{pmatrix}
-1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

Finally, we can combine these two matrices:

\[
\begin{pmatrix}
-1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}\begin{pmatrix}
1 & 0 & 0 & -2 \\
0 & 1 & 0 & -3 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix} = \begin{pmatrix}
-1 & 0 & 0 & 2 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & -3 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]
We can check that this is right by applying it to some sample points. For example, applying this matrix to \((2,3,0)\) produces \((0,0,0)\), which is right because this is the new origin. Applying it to \((2,4,0)\) produces \((0,0,1)\), which is right, because this point is one unit above the viewer. The viewer is looking up, so this point appears to be one unit in front of her. Similarly, the point \((2,3,1)\) should appear to be one unit above her because the \(z\) direction is now up to her, and indeed this point does turn up at \((0,1,0)\).

3. Projection

Suppose we have a camera with a focal point at \((0,0,0)\), and an image plane of \(z=1\). We have two lines. Line A has vertices \((0,0,2)\) and \((7,0,2)\). Line B has vertices at \((0,0,2)\) and \((7,0,12)\).

(a) If we project these lines with perspective projection, which statement is true: (i) Line A appears longer than line B in the image; (ii) Line B appears longer than line A in the image; (iii) The lines appear to be the same length in the image.

In this setup, \((x,y,z)\) projects to \((x/z, y/z)\). So in the image, line A has vertices \((0,0)\), \((7/2,0)\), while line B has vertices \((0,0)\), \((7/12,0)\). Clearly A appears longer.

(b) If we project these lines with orthographic projection, which statement is true: (i) Line A appears longer than line B in the image; (ii) Line B appears longer than line A in the image; (iii) The lines appear to be the same length in the image.

With orthographic projection we just need to eliminate the \(z\) coordinate of each point. So both lines have the same vertices, \((0,0)\), \((7,0)\), and the same length.

(c) If we move the focal point of the camera, might your answer to (a) and (b) change? Explain.

If we change the focal point this will have no effect orthographic projection. We project in parallel into the image plane, and no focal point is used.

With perspective, shifting the focal point in the \(z\) direction has the effect of scaling the whole image, which does not change the relative size of the lines. However, shifting the focal point in the \(x\) or \(y\) direction changes the amount to which each line is foreshortened. If the new focal point is at \((7,0,z)\) for any \(z < 1\), then it will be collinear with the endpoints of the two line segments, and they'll appear to have the same length. If \(x\) becomes bigger than 7, line B will appear longer, as illustrated below.
4. Digitization

(a) Suppose there is a polygon in an image. We begin at point p, which is not on the boundary of the polygon. We move 20 pixels up, turn 90 degrees to the right, move 20 pixels, turn 90 degrees to the right, and continue until we leave the image. If we cross the polygon boundary three times, did we (a) begin inside the polygon; (b) begin outside the polygon; (c) can’t tell from this information.

(b) Suppose in the last turn we turn, instead of turning right we turn 180 degrees and go backwards until we leave the image. If we cross the polygon boundary three times, did we (a) begin inside the polygon; (b) begin outside the polygon; (c) can’t tell from this information.

Every time we cross the boundary of a polygon we either go from inside to outside or outside to inside. This is true no matter what path we take. So if we cross three times, and wind up outside, we must have started inside.

5. Texture

Two textures were made using Derek’s implementation of Perlin noise. Below we show a plot of the intensity of each texture along a horizontal line. What is the difference between them? That is, what feature of the program was used to make them look different, and how was it used?
The plot on the left is smoother, with many fewer local maxima and minima. Adding more octaves gives the curve more high frequencies, and makes it look rougher. This was what was done on the right. Turbulence could also make the plot rougher, but it does this by adding discontinuities at only one horizontal line.

Some students suggested that the rougher curve occurred due to the use of linear interpolation. This doesn’t exactly fit the question statement, but it’s a reasonable theory.

6. Color

We describe colors using RGB.

(a) Which is more saturated, (.2, .3, .6) or (.4, .5, .8)?
(b) Do (.2, .3, .4) and (.1, .2, .3) have the same hue or different hue?
(c) Which is more saturated: (.3, .7, .4) or (.8, .4, .5)?
(d) Do (.5, .3, .7) and (.6, .2, 1) have the same or different hue?
(e) Which has a higher value, (.2, .3, .6) or (.4, .5, .8)?

We can answer these questions if we know the following things: 1) Adding gray to a color makes it less saturated; 2) Shuffling the RGB values doesn’t change the saturation; 3) Adding gray to a color does not change the hue; 4) scaling all the RGB values by the same constant does not change the hue; 5) Adding gray adds light, which increases the value.

(a) (.2,.3,.6) + (.2,.2,.2) = (.4,.5,.8), so (.4,.5,.8) is more saturated.
(b) (.2,.3,.4) = (.1,.1,.1)+(.1,.2,.3) so they have the same hue.
(c) (.8,.4,.5) is just as saturated as (.4,.8,.5).
 (.3,.7,.4) = (.3,.3,.3) + (0,.4,.1). (.4,.8,.5) = (.4,.4,.4) + (0,.4,.1). The two colors are the same thing with more gray added to (.4,.8,.5), so (.3,.7,.4) is more saturated than (.4,.8,.5), and so it is more saturated than (.8,.4,.5).
(d) \((.5,.3,.7) = (.3,.3,.3) + (.2,0,.4)\). \((.6,.2,1) = 2\times(.2,0,.4) + (.2,.2,.2)\). So they have the same hue.
(e) \((.2,.3,.6) + (.2,.2,.2) = (.4,.5,.8)\), so \((.4,.5,.8)\) has higher value.

7. **Challenge problem for extra credit:** Suppose we have a perspective camera with a focal point at \((0,0,0)\) and an image plane at \(z=1\). Can you give an example of a right triangle that projects into the image to a triangle containing an angle greater than 90 degrees? Either provide an example, or show why this is not possible. If you provide an example, give coordinates of the triangle vertices before and after projection.

One way to solve this is to notice that with a right triangle, the length of the hypotenuse is equal to the sum of the squares of the other sides. If one side has a squared length that is even more than the sum of the squares of the other sides, then the angle opposite it will be even more than 90 degrees. So we pick a viewpoint that will make the hypotenuse longer relative to the other two sides. We accomplish this by making the hypotenuse close to the image plane, and the other sides farther. For example, suppose the hypotenuse has one vertex at \((0,0,1)\) and the other at \((x,0,1)\) for some \(x\). We’ll pick the other vertex farther away, say at \((2,2,2)\). Then for the Pythagorean theorem to hold, we need \(9 + (x-2)^2 + 4 + 1 = x^2\). \(9 - 4x + 4 + 4 + 1 = 0\). \(x = 9/2\). So in 3D, the triangle forms a right angle.

When we project these points to the image, they appear at \((0,0), (1,1), (9/2,0)\), and the vertex at \((1,1)\) has an angle greater than 90 degrees.