



Lesson 27: Designing Your Own Game

Student Outcomes

- Students will create a short animation in ALICE 3.1 using one-step procedures.
- Students will apply their understanding of the mathematics of projecting three-dimensional images onto two-dimensional surfaces by writing matrices to represent the motions in the animations they create in ALICE.

Lesson Notes

This is the last lesson in this topic and this module. Lesson 27 is a culminating lesson where students are able to see how their work with functions, matrices, vectors, and linear transformations all come together.

Prior to this lesson, the students should have become familiar with the ALICE program, having opportunities to explore it in previous homework assignments. The ALICE 3.1 program should be available to students during the lesson. Students will use it to create a brief scene including one-step turn and roll procedures. They will describe the motions of the ALICE characters with respect to the scene and also describe them as rotations about specific axes. Then students will apply the mathematics from the previous lesson to use matrices to represent the motions of the characters in their scenes.

If there is only one copy of ALICE available, this lesson could be completed by having the class create one collective animation. Selected students could choose the setting and characters. Then, volunteers could each apply a one-step roll or turn procedure, while the rest of the students determine a matrix operation that would appropriately represent the movement. The students could be prompted to create distinct procedures, so the movements represent rotations of various degrees and around various axes. The activities and procedures addressed in this lesson will provide opportunities for students to reason abstractly about the movements of the animated objects (MP.2) and to model the movements of objects using matrix operations (MP.4).

Classwork

Discussion (10 minutes): Reviewing Projections of 3D objects into 2D space

Through a teacher-led series of questions, the students will review what they have learned about projecting three-dimensional objects into two-dimensional space, including using matrices to represent projected images and rotations of points. If technology is not available, the question addressing the ALICE program could be skipped.

- In a sentence, explain the function of the ALICE program to a person new to the program.
 - *It creates 3D animations using procedures to manipulate objects within the ALICE world.*
- What is actually happening to create the illusion of three-dimensional objects in a two-dimensional space?
 - *The points composing the three-dimensional objects are projected onto a two-dimensional screen.*
- And how is this projection formed?
 - *It is the intersection of the viewer's line of sight to the real object with the flat surface onto which it is projected.*

- How can this type of projection be drawn by hand?
 - *Answers may vary but should include using a horizon line and vanishing points.*
- And will all projections of a point look exactly the same?
 - *No, they will vary based on the placement of the horizon point and the vanishing points.*
- We attempted this process in Lesson 25 by projecting a cube onto a flat screen. What challenges did we encounter?
 - *Answers may vary but could include difficulties in seeing the points of intersection between the line of sight and the screen prior to the addition of the horizon line and vanishing points.*
- What might be some additional challenges we would face in trying to project more complex three-dimensional images onto a screen by hand?
 - *Answers may vary and could include the need for projecting numerous points to get an accurate projection, which could be time consuming and cumbersome when done by hand.*
- How can mathematics be applied to expedite the process of projecting three-dimensional objects into two-dimensional space?
 - *We can set the viewer's eye at the origin and write an equation from the eye to each point to be projected. The image point on the screen will be located on that line at the intersection with the screen.*
- So if we have the point $A = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$, how can we represent the line through the viewer's eye and point A parametrically?
 - *The equation of the line can be represented as $\begin{bmatrix} ta_1 \\ ta_2 \\ ta_3 \end{bmatrix}$, where t is a real number.*
- And what does this line represent?
 - *It is a dilation of the point $\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$ about the origin.*
- Let's say we are projecting our point A onto a screen defined by the plane $y = 1$. How could we find the coordinates of the point?
 - *Since $y = 1$, we need to find a value t such that $ta_2 = 1$. Therefore, $t = \frac{1}{a_2}$, and the image point is*

$$\begin{bmatrix} ta_1 \\ ta_2 \\ ta_3 \end{bmatrix} = \begin{bmatrix} \frac{a_1}{a_2} \\ 1 \\ \frac{a_3}{a_2} \end{bmatrix}.$$
- How would our projected point change if we chose the screen to be at $y = 2$?
 - *We would need to find the value t so that $ta_2 = 2$, but otherwise, the procedure would be the same.*
- How would our projected point change if we chose the screen to be at $z = 1$?
 - *We would need to find the value t so that $ta_3 = 1$, but otherwise, the procedure would be the same.*
- Let's say we have created a seafloor scene in ALICE. In the scene we placed a shark whose pivot point is represented by the coordinates $(3, 5, 0.2)$. What would be the coordinates of this point projected onto a screen at $y = 1$? At $z = 2$?
 - *For $y = 1$, $5t = 1$, so $t = 0.2$, and the projected image point is $(0.6, 1.0, 0.04)$.*
 - *For $z = 2$, $0.2t = 2$, so $t = 10$, and the projected image point is $(30, 50, 2)$.*

- Now let's discuss rotating points in three-dimensional space. How can you use matrices to represent the rotation of a point about the z -axis through an angle θ ?

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

- How can you use matrices to represent the rotation of a point about the y -axis through an angle θ ?

$$\begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

- And a rotation about the x -axis through an angle θ ?

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

- And what are the coordinates of point A rotated about the x -axis?

$$\begin{bmatrix} a_1 \\ a_2 \cos(\theta) - a_3 \sin(\theta) \\ a_2 \sin(\theta) + a_3 \cos(\theta) \end{bmatrix}$$

- How would we project these points onto the plane $y = 1$?

$$\text{Multiply the rotation matrix by } t \text{ so that } t(a_2 \cos(\theta) - a_3 \sin(\theta)) = 1.$$

- And what would that matrix look like?

$$\text{If } t(a_2 \cos(\theta) - a_3 \sin(\theta)) = 1, \text{ then } t = \frac{1}{a_2 \cos(\theta) - a_3 \sin(\theta)}, \text{ and the projected points are represented by}$$

$$\begin{bmatrix} \frac{a_1}{a_2 \cos(\theta) - a_3 \sin(\theta)} \\ 1 \\ \frac{a_2 \sin(\theta) + a_3 \cos(\theta)}{a_2 \cos(\theta) - a_3 \sin(\theta)} \end{bmatrix}.$$

Example 1 (10 minutes)

Open the ALICE program, and demonstrate how it works by completing Example 1. As students observe the setup of a simple scene with one character and the manipulation of the character within the ALICE world, they will be reminded of how to create and manipulate objects in ALICE, which should help them as they create their own scene with a partner in the exercises that follow. While presenting the example, use matrix operations to represent the motions of characters. If technology is not available, the students could work in pairs or small groups to create an idea for a 3D animation. They could create a scene with multiple characters and determine a few motions that each character would make in their scene. These motions could be described using captioned sketches, or the students could create paper characters, which could be manipulated to demonstrate their motions. Parameters for the motions of the characters could be set (e.g., between 2–4 characters per scene; 1–2 movements per character; at least two rotations around different axes must be represented). The scenes could be presented, and students who are not presenting their scene could use matrix operations to represent the movements of the characters (presentations discussed in the next lesson section).

Scaffolding:

Advanced students could represent multiple movements using matrix operations, such as multiple rotations or a rotation followed by a translation.

- Once we have selected the dolphin, I am going to move it left and right. Look at the coordinates of the dolphin on the right side of the screen. What coordinate is changing, and how is it changing?
 - *The x-coordinate changes when the dolphin moves left and right. Moving to the left increases the value of x, and moving to the right decreases it.*
- So our positive x-axis is to the left from the viewer's vantage point when the dolphin is facing with its nose towards the viewer. Now, if I click on the arrow pointing up, I can move the dolphin up and down. What coordinate is changing now, and how is it changing?
 - *The y-coordinate changes when the dolphin moves up and down, with the up direction increasing the value of y.*
- So our y-axis is vertical when the dolphin is facing forward with its back to the sky. How about when we move the dolphin forward and backward?
 - *The z-coordinate changes. It increases as the dolphin moves back.*
- Now we have moved the dolphin so its pivot point is at the origin. How can we represent translations of the dolphin using matrix operations?
 - *Answers should address adding the matrix $\begin{bmatrix} a \\ b \\ c \end{bmatrix}$ to $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$, where a represents the number of units the dolphin is translated to the left, b represents the number of units the dolphin is translated up, and c represents the number of units the dolphin is translated backward.*
- Now we have selected the rotation button and selected the dolphin. What does the dolphin appear to do when we rotate it about the x-axis?
 - *A forward or backward somersault.*
- And the y-axis?
 - *The dolphin turns in the water.*
- And the z-axis?
 - *The dolphin rolls on its side.*
- And how can we generalize the rotations of the dolphin about the axes using matrices?
 - *Rotation about x-axis:*

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(c) & -\sin(c) \\ 0 & \sin(c) & \cos(c) \end{bmatrix}$$
where c is the degrees of rotation counterclockwise about the axis
 - *Rotation about y-axis:*

$$\begin{bmatrix} \cos(c) & 0 & -\sin(c) \\ 0 & 1 & 0 \\ \sin(c) & 0 & \cos(c) \end{bmatrix}$$
where c is the degrees of rotation counterclockwise about the axis
 - *Rotation about z-axis:*

$$\begin{bmatrix} \cos(c) & -\sin(c) & 0 \\ \sin(c) & \cos(c) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
where c is the degrees of rotation counterclockwise about the axis
- Now we have selected the procedure: this.dolphin.roll RIGHT 0.25 as seen by this duration 2.0 BEGIN_AND_END_ABRUPTLY. What happened when we ran the program?
 - *The dolphin does a quarter backwards somersault out of the water in a counterclockwise rotation about the negative z-axis.*

- How does this make sense based on the procedure we created?
 - *Answers will vary but should address that in the default orientation, a roll to its right would appear to the viewer as a roll onto its left side, which is a counterclockwise motion.*
- Now you will get to use what we have reviewed together to create your own animation and to describe the motions using matrices.

Example 1

- a. Select the Sea Surface as your background scene, and select the setup scene mode. From the Gallery By Class Hierarchy, select Swimmer Classes, then Marine Mammals, and a Dolphin. Name your dolphin. Describe what you see.

We have a scene with an ocean surface and a dolphin facing forward in the center.

- b. Click on the dolphin, and from the handle style buttons in the top right corner of the screen, select translation. By selecting the arrows on the dolphin, we can move it to different locations in the screen. Move the dolphin left and right, up and down, forward and backward. Then, move it so that its coordinates are $(0, 0, 0)$.

- c. Use a matrix to describe each of the movements of the dolphin from its location at the origin.

- i. Move 2 units right.

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -2 \\ 0 \\ 0 \end{bmatrix}$$

- ii. Move 4 units down.

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -4 \\ 0 \end{bmatrix}$$

- iii. Move 3 units forward.

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -3 \end{bmatrix}$$

- d. Click on the rotation button from the handle style buttons, and practice rotating the dolphin about the three axes through its center. Use a matrix to represent the motion of the dolphin described. Assume the center of rotation of the dolphin is at the origin.

- i. Rotation counterclockwise one full turn about the z-axis

$$\begin{bmatrix} \cos(360) & -\sin(360) & 0 \\ \sin(360) & \cos(360) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- ii. Rotation counterclockwise one half turn about the x-axis

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(180) & -\sin(180) \\ 0 & \sin(180) & \cos(180) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Scaffolding:

- Students who feel comfortable in ALICE can proceed to creating their scene without the tutorial.
- Struggling students may benefit from a written set of directions.

- e. Select Edit Code from the screen. Drag and drop this.dolphinturn from the procedures menu and drop it into the declare procedures region on the right. Select from the drop-down menus: this.dolphin turn LEFT 0.25 as seen by this duration 2.0 BEGIN_AND_END_ABRUPTLY. Then run the program. Describe what you see. Represent the motion using a matrix.

The dolphin undergoes a quarter counterclockwise rotation in a horizontal circle about the y-axis.

$$\begin{bmatrix} \cos(90) & 0 & -\sin(90) \\ 0 & 1 & 0 \\ \sin(90) & 0 & \cos(90) \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

- f. Drag and drop this.dolphinroll from the procedures menu, and drop it into the declare procedures region on the right beneath the turn procedure. Select from the drop-down menus: this.dolphin roll RIGHT 3.0 as seen by this duration 2.0 BEGIN_AND_END_ABRUPTLY. Then run the program. Describe what you see. Represent the motion using a matrix.

The dolphin performs three backwards counterclockwise somersaults in the vertical plane, about the negative z-axis.

$$\begin{bmatrix} \cos(1080) & -\sin(1080) & 0 \\ \sin(1080) & \cos(1080) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Exercises (15 minutes)

Give students time to work on the problems below in ALICE. If it is possible, the students could complete the tasks in pairs. Once they have completed the exercises, they can consult with another pair to verify their solutions. Select students to present their scenes and solutions to the class as time permits; for example, perhaps each pair could demonstrate one procedure while the pair with whom they partnered displays the matrix that represents the motion in the 3D ALICE world, as well as on a projected screen. Some pairs of students may need some one-on-one assistance either setting up the scene or representing the projecting points using matrices. When students do pivots, have them assume the pivot point is at the origin. If technology is not available, continue the activity begun earlier in the lesson, where students present their ideas for animations, including descriptions or demonstrations of character movements. Have the students who are not presenting represent the movements of each character using matrix operations. These students could share their matrices and justify their responses, while other students could provide supporting or refuting evidence.

Exercises

1. Open ALICE 3.1. Select a background and 3 characters to create a scene. Describe the scene, including the coordinates of the pivot point for each character and the direction each character is facing.

Scene description: Answers will vary. I created a sea surface scene that contains an adult walrus facing forward, a baby walrus to the left of the adult and facing it (looking to the right), and a dolphin breaching on its back, with its nose pointed up and to the right. The dolphin is behind and to the left of the walruses.

Coordinates of the pivot point for each character:

Answers will vary. For example, dolphin (4.16, 0.5, 4.1); baby walrus (0.6, 0.11, 0.55); adult walrus (-0.46, -0.03, 1.11).

- 2.

- a. Describe the location of a plane $x = 5$ in your scene from the perspective of the viewer.

The plane $x = 5$ is a vertical plane 5 units to the left of the center of the screen.

- b. Determine the coordinates of the pivot points for each character if they were projected onto the plane $x = 5$.

Answers will vary. An example of an acceptable response is shown.

For the dolphin, $4.16t = 5$, so $t \approx 1.20$ and image = $\begin{bmatrix} 5 \\ 6 \\ 4.92 \end{bmatrix}$.

For the baby walrus, $0.6t = 5$, so $t = \frac{25}{3}$ and image = $\begin{bmatrix} 5 \\ 0.92 \\ 4.58 \end{bmatrix}$.

For the adult walrus, $-0.46t = 5$, so $t \approx -10.87$ and image = $\begin{bmatrix} 5 \\ 0.33 \\ -12.07 \end{bmatrix}$.

3. Create a short scene that includes a one-step turn or roll procedure for each of the characters. The procedures should be unique. Write down the procedures in the space provided. After each procedure, describe what the character did in the context of the scene. Then describe the character's motion using transformational language. Finally, represent each procedure using matrix operations.

Answers will vary. An example of an acceptable response is shown.

Procedure 1—adult walrus: this.adult walrus TURN forward 4.0 as seen by this duration 2.0.

Character motion (in scene): The adult walrus performed four forward somersaults toward the foreground.

Transformational motion: The adult walrus performed a counterclockwise rotation of four full turns about the x -axis.

Matrix representation of motion: $\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(1440) & -\sin(1440) \\ 0 & \sin(1440) & \cos(1440) \end{bmatrix} A$ for any point A on the walrus translated to be centered around the origin.

Procedure 2—baby walrus: this.babywalrus TURN left 2.0 as seen by this duration 2.0.

Character motion (in scene): The baby walrus spun twice in a counterclockwise circle next to the adult.

Transformational motion: The baby walrus rotated about the y -axis counterclockwise.

Matrix representation of motion: $\begin{bmatrix} \cos(720) & 0 & -\sin(720) \\ 0 & 1 & 0 \\ \sin(720) & 0 & \cos(720) \end{bmatrix} B$ for any point B on the walrus translated to be centered around the origin.

Procedure 3—Dolphin: this.dolphin ROLL left 4.0 as seen by this duration 2.0.

Character motion (in scene): The dolphin performed four back somersaults out of the water.

Transformational motion: The dolphin rotated counterclockwise about the z -axis.

Matrix representation of motion:

$\begin{bmatrix} \cos(720) & -\sin(720) & 0 \\ \sin(720) & \cos(720) & 0 \\ 0 & 0 & 1 \end{bmatrix} C$ for any point C on the dolphin translated to be centered around the origin.

Closing (5 minutes)

Have students respond in writing to the prompt. After a few minutes, several students should be selected to share their thoughts. If technology is not available, the students can respond to the prompt with respect to rotations of 3D objects in any 3D animation situation.

- Make a generalization about how one-step turns or rolls in ALICE can be represented using matrix operations. Use an example from your animation to support your response.

- *Answers will vary. An example of an acceptable response is included:*

One-step turns and rolls can be represented as rotations of a given point on the character once it has been translated to the origin. This rotation can be represented using a rotation matrix applied to the translated point A about the axis of rotation for a rotation of θ degrees. For example, the adult walrus performed a rotation about the x -axis, which can be represented as

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(1440) & -\sin(1440) \\ 0 & \sin(1440) & \cos(1440) \end{bmatrix} A$$

Exit Ticket (5 minutes)

Name _____

Date _____

Lesson 27: Designing Your Own Game

Exit Ticket

Consider the following set of code for the ALICE program featuring a bluebird whose center is located at $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$.

```
this.bluebird resize 2.0  
this.bluebird resizeHeight 0.5  
this.bluebird turn RIGHT 0.25  
this.bluebird move FORWARD 1.0
```

For an arbitrary point x on this bluebird, write the four matrices that represent the code above, and state where the point ends after the program runs.

Exit Ticket Sample Solutions

Consider the following set of code for the ALICE program featuring a bluebird whose center is located at $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$.

```
this.bluebird.resize 2.0
this.bluebird.resizeHeight 0.5
this.bluebird.turn RIGHT 0.25
this.bluebird.move FORWARD 1.0
```

For an arbitrary point x on this bluebird, write the four matrices that represent the code above, and state where the point ends after the program runs.

Resize 2.0: $\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$

ResizeHeight 0.5: $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Turn RIGHT 0.25: turning right is a rotation about the y -axis clockwise, so we get

$$\begin{bmatrix} \cos(-90) & 0 & -\sin(-90) \\ 0 & 1 & 0 \\ \sin(-90) & 0 & \cos(90) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

Move forward is a translation by 1 unit in the positive x direction after the turn right. There is no linear transformation

we can use, but we can write this as $x' + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$.

x would end at

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

If $x = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$, then the image of x at the end of the program would be $\begin{bmatrix} 2c + 1 \\ b \\ -2a \end{bmatrix}$.

Problem Set Sample Solutions

1. For the following commands, describe a matrix you can use to get the desired result. Assume the character is centered at the origin, facing in the negative z direction with positive x on its right for each command. Let c be a nonzero real number.

- a. move LEFT c

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -c \\ 0 \\ 0 \end{bmatrix}$$

- b. move RIGHT c

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} c \\ 0 \\ 0 \end{bmatrix}$$

c. move UP c

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ c \\ 0 \end{bmatrix}$$

d. move DOWN c

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -c \\ 0 \end{bmatrix}$$

e. move FORWARD c

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -c \end{bmatrix}$$

f. move BACKWARD c

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix}$$

g. turn LEFT c

$$\begin{bmatrix} \cos(360c) & 0 & -\sin(360c) \\ 0 & 1 & 0 \\ \sin(360c) & 0 & \cos(360c) \end{bmatrix}$$

h. turn RIGHT c

$$\begin{bmatrix} \cos(-360c) & 0 & -\sin(-360c) \\ 0 & 1 & 0 \\ \sin(-360c) & 0 & \cos(-360c) \end{bmatrix}$$

i. turn FORWARD c

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-360c) & -\sin(-360c) \\ 0 & \sin(-360c) & \cos(-360c) \end{bmatrix}$$

j. turn BACKWARD c

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(360c) & -\sin(360c) \\ 0 & \sin(360c) & \cos(360c) \end{bmatrix}$$

k. roll LEFT c

$$\begin{bmatrix} \cos(-360c) & -\sin(-360c) & 0 \\ \sin(-360c) & \cos(-360c) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

l. roll RIGHT c

$$\begin{bmatrix} \cos(360c) & -\sin(360c) & 0 \\ \sin(360c) & \cos(360c) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

m. resize c

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

n. resizeWidth c

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

o. resizeHEIGHT c

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

p. resizeDEPTH c

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

2. In each of the transformations above, we have assumed that each animation will take one second of time. If T is the number of seconds an animation takes and t is the current running time of the animation, then rewrite the following commands as a function of t .

a. move RIGHT c duration T

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} c \\ 0 \\ 0 \end{bmatrix} \frac{t}{T}$$

b. turn FORWARD c duration T

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-360c) & -\sin(-360c) \\ 0 & \sin(-360c) & \cos(-360c) \end{bmatrix} \frac{t}{T}$$

c. resize c duration T

$$\begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & c \end{bmatrix} \frac{t}{T}$$

3. For computational simplicity, we have been assuming that the pivot points of our characters occur at the origin.

- a. If we apply a rotation matrix like we have been when the pivot point is at the origin, what will happen to characters that are not located around the origin?

The characters will trace out a circular path much like we saw in previous lessons when we were rotating points instead of objects.

- b. Let $x = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ be the pivot point of any three-dimensional object and $A = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ a point on the surface of the object. Moving the pivot point to the origin has what effect on A ? Find A' , the image of A after moving the object so that its pivot point is the origin.

$$\begin{aligned} A' &= A - x \\ &= \begin{bmatrix} a \\ b \\ c \end{bmatrix} - \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\ &= \begin{bmatrix} a - x \\ b - y \\ c - z \end{bmatrix} \end{aligned}$$

- c. Apply a rotation of θ about the x -axis to A' . Does this transformation cause a pivot or what you described in part (a)?

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} a - x \\ b - y \\ c - z \end{bmatrix} = \begin{bmatrix} a - x \\ \cos(\theta)(b - y) - \sin(\theta)(c - z) \\ \sin(\theta)(b - y) + \cos(\theta)(c - z) \end{bmatrix}$$

Since the pivot point has been shifted to the origin, the rotation rotates the object about itself instead of through space.

- d. After applying the rotation, translate the object so that its pivot point returns to $x = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$. Was the x -coordinate affected by the rotation? Was the pivot point?

$$\begin{bmatrix} a - x \\ \cos(\theta)(b - y) - \sin(\theta)(c - z) \\ \sin(\theta)(b - y) + \cos(\theta)(c - z) \end{bmatrix} + \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a \\ \cos(\theta)(b - y) - \sin(\theta)(c - z) + y \\ \sin(\theta)(b - y) + \cos(\theta)(c - z) + z \end{bmatrix}$$

No, the x -coordinate stayed the same once we shifted it back to the starting point. The pivot point would also have stayed the same since we shifted it to the origin, the origin does not change during rotations, and then shifted it back in the opposite direction.

- e. Summarize what you found in (a)–(d).

To pivot points that are not centered on the origin, we have to translate the object so that its pivot point is at the origin, then perform the rotation, and then shift the object back. If R is the rotation matrix, A is the point on the surface of the object, and x is the pivot point, then we do the following:

$$R(A - x) + x$$

for each point A on the surface of the object.

Extension:

4. In first-person computer games, we think of the camera moving left-right, forward-backward, and up-down. For computational simplicity, the camera and screen stays fixed and the objects in the game world move in the opposite direction instead. Let the camera be located at $(0, 0, 0)$, the screen be located at $z = 1$, and the point $v = \begin{bmatrix} 10 \\ 6 \\ 5 \end{bmatrix}$ represent the center of an object in the game world. If a character in ALICE is the camera, then answer the following questions.

- a. What are the coordinates of the projection of v on the screen?

$$v' = \begin{bmatrix} 2 \\ 1.2 \\ 1 \end{bmatrix}$$

- b. What is the value of the image of v as the character moves 4 units closer to v in the x direction?

$$v' = \begin{bmatrix} 6 \\ 6 \\ 5 \end{bmatrix}$$

- c. What are the coordinates of the projection of v' ?

$$v' = \begin{bmatrix} 1.2 \\ 1.2 \\ 1 \end{bmatrix}$$

- d. If the character jumps up 6 units, then where does the image of v move on the screen?

v moves to the apparent position $\begin{bmatrix} 10 \\ 0 \\ 5 \end{bmatrix}$, which appears to be located at $\begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}$ on the screen.

- e. In Lesson 26, you learned that if the camera is not in the standard orientation, that rotation matrices need to be applied to the camera first. In Lesson 27, you learned that rotation matrices only pivot an object if that object is located at the origin. If a camera is in a non-standard orientation and not located at the origin, then should the rotation matrices be applied first or a translation to the origin?

The translation to the origin needs to take place before the rotation or else the camera will be tracing out a circular path around the axes instead of rotating to the correct position.