As the Cube Turns

Programming and Transformational Geometry

Teacher's Guide

This material is based upon work supported by the National Science Foundation under award numbers ESI-9255262, ESI-0137805, and ESI-0627821. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

It's About Time
333 North Bedford Road
Mount Kisco, NY 10549
www.IAT.com

First Edition Authors
Dan Fendel, Diane Resek, Lynne Alper, and Sherry Fraser

Contributors to the Second Edition
Sherry Fraser, Jean Klanica, Brian Lawler, Eric Robinson, Lew Romagnano, Rick Marks, Dan Brutlag, Alan Olds, Mike Bryant, Jeri P. Philbrick, Lori Green, Matt Bremer, Margaret DeArmond

Interactive Mathematics Program Year 4 Publishing Team

Project Editor
Josephine Noah

IMP Second Edition Publishing Team

Project Editor
Sharon Taylor

Consulting Editor
Mali Apple

Project Administrator
Juliana Tringali

Professional Reviewer
Rick Marks, Sonoma State University

Calculator Materials Editor
Josephine Noah

Math Checker
Carrie Gongaware

Production Editor
Andrew Jones

Contents

Introduction

As the Cube Turns Unit Overview vi

As the Cube Turns and the Common Core State Standards for Mathematics ix

Pacing Guides x

Materials and Supplies xviii

Assessing Progress xix

Supplemental Activities overview xxii

Activity Notes

Calculator Pictures 1

Picture This! 2

POW 7: "A Sticky Gum Problem" Revisited 7

Starting Sticky Gum 8

Programming Without a Calculator 10

Programming Loops 14

Learning the Loops 15

An Animated Shape 22

A Flip Book 26

Movin' On 28

Some Back and Forth 31

Arrow 33

Sunrise 34

POW 8: A Wider Windshield Wiper, Please 36

Translation in Two Dimensions 38

Move That Line! 39

Double Dotting 47

Memories of Matrices 48

Cornering the Cabbage 52

Rotation in Two Dimensions 56

Goin' Round the Origin 58

Double Trouble 64
The Sine of a Sum 67
A Broken Button 70
Oh, Say What You Can See 73
Comin' Round Again (and Again…) 75
More Memories of Matrices 81
Taking Steps 86
How Did We Get Here? 88
Swing That Line! 90
Doubles and Differences 93
What's Going On Here? 95
Projecting Pictures 98
From Three Dimensions to Two 100
POW 9: An Animated POW 102
"A Snack in the Middle" Revisted 105
Fractional Snacks 109
More Walking for Clyde 113
Monorail Delivery 115
Another Mystery 118
A Return to the Third Dimension 120
Where's Madie? 124
And Fred Brings the Lunch 126
Flipping Points 129
Where's Bonita? 131
Lunch in the Window 133
Further Flips 137
Cube on a Screen 139
Spiders and Cubes 144
Find Those Corners! 146
An Animated Outline 155
Mirrors in Space 156
Where Are We Now? 158
Rotation in Three Dimensions 159

Follow That Point! 160
One Turn of a Cube 163
Rotation Matrix in Three Dimensions 165
The Turning Cube Outline 167
Beginning Portfolio Selection 170
Creating Animations 171
An Animated POW Write-up 172
Continued Portfolio Selection 173
As The Cube Turns Portfolio 174

Blackline Masters

Learning the Loops 176
Oh, Say What You Can See 177
"A Snack in the Middle" Revisited 178
A Return to the Third Dimension 180
¼-Inch Graph Paper 181
1-Centimeter Graph Paper 182
1-Inch Graph Paper 183
In-Class Assessment 184
Take Home Assessment 186

Calculator Guide and **Calculator Notes**

Calculator Guide 187
Calculator Notes 215

As The Cube Turns Overview

Intent

In this unit, students use computer programming with a combination of matrix arithmetic, trigonometric identities, and geometric translations, rotations, and projections in two and three dimensions to create an animation of a rotating cube.

Mathematics

The main concepts and skills that students will encounter and practice during the unit are summarized below.

Coordinate Geometry

- Expressing geometric transformations—translations, rotations, and reflections—in terms of coordinates in two and three dimensions
- Finding coordinates a fractional distance along a line segment in two and three dimensions
- Reviewing graphing in three dimensions
- Finding the projection of a point onto a plane from the perspective of a fixed point and developing an algebraic description of the projection process
- Studying the effect of change of viewpoint on projections
- Reviewing polar coordinates

Matrices

- Reviewing the algebra of matrices
- Using matrices to express geometric transformations in two and three dimensions

Programming

- Learning to use a technical manual
- Using loops in programming
- Understanding programs from their code
- Designing and programming animations

Synthetic Geometry and Trigonometry

- Reviewing formulas relating the sine of an angle to the cosine of a related angle
- Deriving the formula for the area of a triangle in terms of the lengths of two sides and the sine of the included angle
- Deriving formulas for the sine and cosine of the negative of an angle
- Deriving formulas for the sine and cosine of the sum of two angles and related variations

Progression

This unit opens with an overhead display, generated by a program on a graphing calculator, of a cube rotating in three-dimensional space. The central unit problem is for students to learn what goes into writing such a program.

In addition to introducing students to such programming issues as the use of loops, this task takes them into several areas of mathematics. They study the fundamental geometric transformations—translations, rotations, and reflections—in two dimensions and express them in terms of coordinates.

Students must also learn about rotations in three dimensions to work on the unit problem. The analysis of rotations builds on their experience (in the Year 3 unit *High Dive* and the Year 4 unit *The Diver Returns*) with trigonometric functions and polar coordinates, and leads them to see the need for and to develop formulas for the sine and cosine of the sum of two angles. The study of these geometric transformations also provides a new setting for students to work with matrices, which they previously studied in connection with systems of linear equations (in the Year 3 unit *Meadows or Malls?*).

Another important component of students' work is the analysis of how to represent a three-dimensional object on a two-dimensional screen. This problem is approached through the use of both physical materials and a metaphorical story of two spiders and the thread that connects them. Students see how projection onto a plane is affected by the choice of the plane and the choice of a viewpoint or center of projection.

The unit closes with a project in which students program an animated graphic display of their own design.

Calculator Pictures: Exploring the graphing calculator's capacity for drawing pictures

Programming Loops: Learning about programming loops, using a For/End type of instruction combination

Translation in Two Dimensions: Expressing two-dimensional translations in terms of coordinates and matrices

Rotation in Two Dimensions: Developing trigonometric formulas and using them to express two-dimensional rotations in terms of coordinates and matrices

Projecting Pictures: Developing the geometric ideas behind projection from a viewpoint onto a plane and expressing projection in terms of coordinates

Rotation in Three Dimensions: Expressing three-dimensional rotations in terms of coordinates and matrices and completing the unit problem

An Animated POW: Completing the animation projects and presenting them to the class, completing the unit assessments, and summing up

Technology Note

The activities work well with the TI-83 Plus or TI-84 Plus calculators, and the TURNCUBE demonstration program is written to run on these two models. We recommend that before using the unit with any other graphing calculator, you confirm that the necessary matrix work and screen-drawing instructions can be carried out on that calculator.

See the Calculator Guide and Calculator Notes for details on how the programming concepts and other calculator ideas in this unit are carried out on the TI-83 Plus or TI-84 Plus.

As the Cube Turns and the Common Core State Standards for Mathematics

IMP is written to address the Common Core State Standards for Mathematics (CCSSM) High School standards.

Standards for Mathematical Practice

The eight Standards for Mathematical Practice are addressed exceptionally well throughout the *IMP* curriculum.

1. Make sense of problems and persevere in solving them.
2. Reason abstractly and quantitatively.
3. Construct viable arguments and critique the reasoning of others.
4. Model with mathematics.
5. Use appropriate tools strategically.
6. Attend to precision.
7. Look for and make use of structure.
8. Look for and express regularity in repeated reasoning.

Standards for Mathematical Content

These content of *As the Cube Turns* is supplemental to the Common Core State Standards, yet reinforces and extends many earlier standards, especially on matrices.

Pacing Guides

50-minute Pacing Guide (38 days)

Day	Activity	In-Class Time Estimate
1	Calculator Pictures	0
	Picture This!	40
	Introduce: *POW 7: "A Sticky Gum Problem" Revisited*	10
	Homework: *Starting Sticky Gum*	0
2	Discussion: *Starting Sticky Gum*	10
	Picture This! (continued)	25
	Homework: *Programming Without a Calculator*	15
3	Discussion: *Programming Without a Calculator*	20
	Programming Loops	0
	Homework: *Learning the Loops*	30
4	Discussion: *Learning the Loops*	20
	An Animated Shape	30
	Homework: *A Flip Book*	0
5	Discussion: *A Flip Book*	10
	An Animated Shape (continued)	40
	Homework: *Movin' On*	0
6	Discussion: *Movin' On*	50
	Homework: *Some Back and Forth*	0
7	Presentations: *POW 7: "A Sticky Gum Problem" Revisited*	10
	Discussion: *Some Back and Forth*	10
	Arrow	20
	Introduce: *POW 8: A Wider Windshield Wiper, Please*	5
	Homework: *Sunrise*	5
8	Discussion: *Sunrise*	20
	Translation in Two Dimensions	0
	Move That Line!	30
	Homework: *Double Dotting*	0

9	Discussion: *Double Dotting*	10
	Move That Line! (continued)	40
	Homework: *Memories of Matrices*	0
10	Discussion: *Memories of Matrices*	50
	Homework: *Cornering the Cabbage*	0
11	Discussion: *Cornering the Cabbage*	30
	Rotation in Two Dimensions	0
	Homework: *Goin' Round the Origin*	20
12	Discussion: *Goin' Round the Origin*	50
	Homework: *Double Trouble*	0
13	Discussion: *Double Trouble*	15
	The Sine of a Sum	35
	Homework: *A Broken Button*	0
14	Discussion: *A Broken Button*	20
	The Sine of a Sum (continued)	30
	Homework: *Oh, Say What You Can See*	0
15	Discussion: *Oh, Say What You Can See*	10
	Homework: *Comin' Round Again (and Again...)*	40
16	*More Memories of Matrices*	50
	Homework: *Taking Steps*	0
17	Discussion: *Taking Steps*	10
	More Memories of Matrices (continued)	35
	Homework: *How Did We Get Here?*	5
18	Discussion: *How Did We Get Here?*	10
	Swing That Line!	35
	Homework: *Doubles and Differences*	5
19	Discussion: *Doubles and Differences*	15
	Swing That Line! (continued)	35
	Homework: *What's Going On Here?*	0
20	Discussion: *What's Going On Here?*	10
	Projecting Pictures	0
	From Three Dimensions to Two (teacher-led discussion)	10
	Presentations: *POW 8: A Wider Windshield Wiper, Please*	15

	Introduce: *POW 9: An Animated POW*	15
	Homework: *"A Snack in the Middle" Revisited*	0
21	Discussion: *"A Snack in the Middle" Revisited*	10
	Fractional Snacks	40
	Homework: *More Walking for Clyde*	0
22	Discussion: *More Walking for Clyde*	10
	Monorail Delivery	40
	Homework: *Another Mystery*	0
23	Discussion: *Another Mystery*	15
	A Return to the Third Dimension	35
	Homework: *Where's Madie?*	0
	Homework: Prepare plans for *POW 9: An Animated POW*?	0
24	Discussion: *Where's Madie?*	10
	And Fred Brings the Lunch	40
	Homework: *Flipping Points*	0
25	Discussion: *Flipping Points*	10
	And Fred Brings the Lunch (continued)	40
	Homework: *Where's Bonita?*	0
26	Discussion: *Where's Bonita?*	10
	Lunch in the Window	40
	Homework: *Further Flips*	0
27	Discussion: *Further Flips*	10
	Cube on a Screen	40
	Homework: *Spiders and Cubes*	0
28	Discussion: *Spiders and Cubes*	10
	Find Those Corners!	40
	Homework: *An Animated Outline*	0
29	*Find Those Corners!* (continued)	50
	Homework: *Mirrors in Space*	0
30	Discussion: *Mirrors in Space*	10
	Find Those Corners! (continued)	40
	Homework: *Where Are We Now?*	0
31	Discussion: *Where Are We Now?*	10
	Rotation in Three Dimensions	0
	Follow That Point!	40

	Homework: *One Turn of a Cube*	0
32	Discussion: *One Turn of a Cube*	10
	Rotation Matrix in Three Dimensions	35
	Homework: *The Turning Cube Outline*	5
33	Discussion: *The Turning Cube Outline*	50
	Homework: *Beginning Portfolio Selection*	0
34	Discussion: *Beginning Portfolio Selection*	10
	An Animated POW	0
	Pairs work on *POW 9: An Animated POW*	40
	Homework: *"An Animated POW" Write-up*	0
35	Presentations: *POW 9: An Animated POW*	50
	Homework: *Continued Portfolio Selection*	0
36	Discussion: *Continued Portfolio Selection*	5
	Presentations: *POW 9: An Animated POW* (continued)	45
	Homework: *"As the Cube Turns" Portfolio*	0
37	In-Class Assessment	40
	Homework: Take-Home Assessment	10
38	Exam Discussion	40
	Unit Reflection	10

90-minute Pacing Guide (25 days)

Day	Activity	In-Class Time Estimate
1	Calculator Pictures	0
	Picture This!	65
	Introduce: *POW 7: "A Sticky Gum Problem" Revisited*	10
	Homework: *Programming Without a Calculator*	15
2	Discussion: *Programming Without a Calculator*	20
	Starting Sticky Gum	40
	Programming Loops	0
	Homework: *Learning the Loops*	30
3	Discussion: *Learning the Loops*	20
	An Animated Shape	70
	Homework: *A Flip Book*	0
4	Discussion: *A Flip Book*	10
	Movin' On	80
	Homework: *Some Back and Forth*	0
5	Discussion: *Some Back and Forth*	10
	Arrow	25
	Translation in Two Dimensions	0
	Move That Line!	50
	Homework: *Sunrise*	5
6	Presentations: *POW 7: "A Sticky Gum Problem" Revisited*	10
	Introduce: *POW 8: A Wider Windshield Wiper, Please*	5
	Discussion: *Sunrise*	20
	Move That Line! (continued)	20
	Double Dotting	35
	Homework: *Memories of Matrices*	0
7	Discussion: *Memories of Matrices*	45
	Rotation in Two Dimensions	0
	Goin' Round the Origin	45

	Homework: *Cornering the Cabbage*	0
8	Discussion: *Cornering the Cabbage*	25
	Goin' Round the Origin (continued)	40
	Double Trouble	25
	Homework: *A Broken Button*	0
9	Discussion: *A Broken Button*	20
	Double Trouble (continued)	15
	The Sine of a Sum	55
	Homework: *Oh, Say What You Can See*	0
10	Discussion: *Oh, Say What You Can See*	10
	Comin' Round Again (and Again...)	75
	Homework: *POW 8: A Wider Windshield Wiper, Please*	5
11	Presentations of *POW 8: A Wider Windshield Wiper, Please*	20
	Introduce: *POW 9: An Animated POW*	20
	More Memories of Matrices	50
	Homework: *Taking Steps*	0
12	Discussion: *Taking Steps*	10
	More Memories of Matrices (continued)	35
	Swing That Line!	40
	Homework: *How Did We Get Here?*	5
13	Discussion: *How Did We Get Here?*	10
	Swing That Line! (continued)	35
	Doubles and Differences	45
	Homework: *What's Going On Here?*	0
14	Discussion: *What's Going On Here?*	10
	Projecting Pictures	0
	From Three Dimensions to Two (teacher-led discussion)	10
	"A Snack in the Middle" Revisited	35
	Fractional Snacks	35
	Homework: Prepare plans for *POW 9: An Animated POW*	0
15	*More Walking for Clyde*	40
	Monorail Delivery	50
	Homework: *Another Mystery*	0

16	Discussion: *Another Mystery*	15
	A Return to the Third Dimension	40
	Where's Madie?	35
	Homework: *Flipping Points*	0
17	Discussion: *Flipping Points*	10
	And Fred Brings the Lunch	80
	Homework: *Where's Bonita?*	0
18	Discussion: *Where's Bonita?*	10
	Lunch in the Window	40
	Cube on a Screen	40
	Homework: *Spiders and Cubes*	0
19	Discussion: *Spiders and Cubes*	10
	Find Those Corners!	80
	Homework: *An Animated Outline*	0
20	*Find Those Corners!* (continued)	30
	Further Flips	35
	Mirrors in Space	25
	Homework: *Where Are We Now?*	0
21	*Mirrors in Space* (continued)	10
	Discussion: *Where Are We Now?*	10
	Rotation in Three Dimensions	0
	Follow That Point!	40
	One Turn of a Cube	30
	Homework: *Beginning Portfolio Selection*	0
22	Discussion: *Beginning Portfolio Selection*	10
	One Turn of a Cube (continued)	10
	Rotation Matrix in Three Dimensions	35
	The Turning Cube Outline	35
	Homework: *Continued Portfolio Selection*	0
	Homework: *"As the Cube Turns" Portfolio*	0
23	*The Turning Cube Outline* (continued)	50
	In-Class Assessment	40
	Homework: Take-Home Assessment	0
24	Exam Discussion	40
	An Animated POW	0

	Pairs work on *POW 9: An Animated POW* Homework: *"An Animated POW" Write-up*	50 0
25	Presentations: *POW 9: An Animated POW* Unit Reflection	80 10

Materials and Supplies

All IMP classrooms should have a set of standard supplies, described in the section "Materials and Supplies for the IMP Classroom" in *A Guide to IMP.* You'll also find a comprehensive list of materials needed for all Year 4 units in the section "Materials and Supplies for Year 4" in the *Year 4 Teacher's Guide* general resources.

Listed here are the supplies needed for this unit. Also available are general and activity-specific blackline masters, for transparencies or for student worksheets, in the "Blackline Masters" section in *As the Cube Turns* Unit Resources.

***As the Cube Turns* Materials**

- Overhead graphing calculator
- Calculator manuals (1 per pair of students)
- Large foam cube (or similar object) and two sticks
- Optional: A centimeter cube and two straws for each group
- Optional: A video on the making of animation features
- Optional: A sample flip book (perhaps one made by a former student)
- Large (5″ ´ 8″ or larger) index cards
- Yarn and scissors
- Optional: String
- Sheets of clear acrylic plastic (Plexiglas) or similar material (1 sheet for each pair of students)
- Pens of three colors for writing on the plastic (3 pens for each pair of students)
- Cubes (1 per pair, at least 2 inches on an edge, preferably made by connecting smaller cubes with different colors at the corners)
- Grid chart paper

More About Supplies

Graph paper is a standard supply for IMP classrooms. Blackline masters of 1-Centimeter Graph Paper, ¼-Inch Graph Paper, and 1-Inch Graph Paper are provided, for you to make copies and transparencies.

Assessing Progress

As the Cube Turns concludes with two formal unit assessments. In addition, there are many opportunities for more informal, ongoing assessments throughout the unit. For more information about assessment and grading, including general information about the end-of-unit assessments and how to use them, consult *A Guide to IMP.*

End-of-Unit Assessments

This unit concludes with in-class and take-home assessments. The in-class assessment is intentionally short so that time pressures will not affect student performance. Students may use graphing calculators and their notes from previous work when they take the assessments. You can download unit assessments from the *As the Cube Turns* Unit Resources.

Ongoing Assessment

One of the primary tasks of the classroom teacher is to assess student learning. Although the assigning of course grades may be part of this process, assessment more broadly includes the daily work of determining how well students understand key ideas and what level of achievement they have attained on key skills, in order to provide the best possible ongoing instructional program for them.

Students' written and oral work provides many opportunities for teachers to gather this information. We make some recommendations here of activities to monitor especially carefully that will give you insight into student progress.

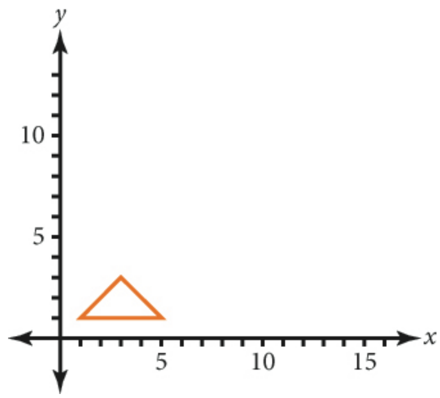
- *Learning the Loops*
- *Move That Line!*
- *Oh, Say What You Can See*
- *Swing That Line!*
- *And Fred Brings the Lunch*
- *Find Those Corners!*
- Work on *POW 9: An Animated POW* (The outline is turned in for *An Animated Outline*, the write-up is turned in for *An Animated POW Write-up*, and presentations are made following that.)

Discussion of Unit Assessments

Have students volunteer to explain their work on each of the problems. Encourage questions and alternate explanations from other students.

In-Class Assessment

Question 1 is a straightforward assessment of whether students understand the matrix notation. The screen might be shown like this:



Students may have used the first triangle in the diagram in Question 2 to confirm the correctness of their results.

For Question 2, the modified program will probably look something like this:

Setup program

Let A be the matrix $\begin{bmatrix} 1 & 1 \\ 3 & 3 \\ 5 & 1 \end{bmatrix}$

Let B be the matrix $\begin{bmatrix} 2 & 1 \\ 2 & 1 \\ 2 & 1 \end{bmatrix}$

For T from 1 to 5

- Draw a line from (a_{11}, a_{12}) to (a_{21}, a_{22})
- Draw a line from (a_{11}, a_{12}) to (a_{31}, a_{32})
- Draw a line from (a_{31}, a_{32}) to (a_{21}, a_{22})
- Replace matrix A by the sum A + B

End the T loop

Out of habit, some students might include "delay" and "clear screen" instructions in their programs. If so, point out that the final screen shows all five triangles, so the "clear screen" instruction should not be included, and the "delay" instruction is not needed.

Question 3 is similar, with each triangle being a 45° counterclockwise rotation around the origin from the previous triangle. The plain-language program might look like the following:

Setup program

Let A be the matrix $\begin{bmatrix} 1 & 1 \\ 3 & 3 \\ 5 & 1 \end{bmatrix}$

Let C be the matrix $\begin{bmatrix} \cos 45° & \sin 45° \\ -\sin 45° & \cos 45° \end{bmatrix}$

For T from 1 to 6

- Draw a line from (a_{11}, a_{12}) to (a_{21}, a_{22})
- Draw a line from (a_{11}, a_{12}) to (a_{31}, a_{32})
- Draw a line from (a_{31}, a_{32}) to (a_{21}, a_{22})
- Replace matrix A by the product $A \cdot C$

End the T loop

Students might use numeric values for sin 45° and cos 45°, both of which are approximately .71.

Take-Home Assessment

For Part I, you might go straight to the explanation (Question 3), because that should also include the answers to Questions 1 and 2. The presenter should explain that $\frac{6-w}{2-w}$ gives the ratio between these two values:

- The difference in z-coordinates between (u, v, w) and the screen
- The difference in z-coordinates between (u, v, w) and the viewpoint

The expression for the numerator shows that the screen has equation $z = 6$, and the expression for the denominator shows that the z-coordinate of the viewpoint is 2.

The presenter should also explain that the expressions $0 - u$ and $1 - v$ give the x- and y-distances from (u, v, w) to the viewpoint, so the x- and y-coordinates of the viewpoint are 0 and 1, respectively. Thus, the viewpoint is at (0, 1, 2).

For Part II, have volunteers share ideas about the development of the rotation formulas. This will probably include the use of polar coordinates to describe the beginning and ending positions of the point and the role of the sine-of-a-sum and cosine-of-a-sum formulas in expressing the result in terms of just x, y, and ϕ.

Supplemental Activities

The unit contains a variety of activities at the end of the student pages that you can use to supplement the regular unit material. These activities fall roughly into two categories.

Reinforcements increase students' understanding and comfort with concepts, techniques, and methods that are discussed in class and are central to the unit.

Extensions allow students to explore ideas beyond those presented in the unit, including generalizations and abstractions of ideas.

The supplemental activities are presented in the teacher's guide and the student book in the approximate sequence in which you might use them. Below are specific recommendations about how each activity might work within the unit. You may wish to use some of these activities, especially the later ones, after the unit is completed.

***Loopy Arithmetic* (extension)** The programming tasks in this activity ask students to make more sophisticated use of For/End loops than required by the unit problem. This activity can be used once students have some basic experience with For/End loops, such as after *Learning the Loops.*

***Sum Tangents* (reinforcement)** This activity follows smoothly from the development of the formulas for the sine and cosine of a sum and can be used anytime after *Comin' Round Again (and Again...)*.

***Moving to the Second Quadrant* (reinforcement)** In *The Sine of a Sum*, students proved the formula for the sine of the sum of two acute angles. They were told that the formula they found works for all angles, but no proof was provided for other angles. In this activity, they prove the formula for the case in which one of the angles is in the second quadrant and the other is in the first quadrant. This activity can be done anytime after *Comin' Round Again (and Again...)*, when the formula for the sine of a sum is applied to find the cosine of a sum. The supplemental activities *Adding 180°* and *Sums for All Quadrants* continue this work.

***Adding 180°* (reinforcement)** This activity is similar to Questions 1 and 2 of the supplemental activity *Moving to the Second Quadrant.* The formula students are asked to develop in Question 1 is needed in the supplemental activity *Sums for All Quadrants.*

***Sums for All Quadrants* (extension)** In this activity, students prove the sine-of-a-sum formula for all angles, using a quadrant-by-quadrant analysis. The activity is a follow-up to the two preceding supplemental activities.

***Polar Complex* (extension)** This activity builds on the development in the unit of the formulas for the sine and cosine of the sum of two angles and can be assigned any time after *Comin' Round Again (and Again...)*. Students examine the topic of representation of complex numbers in polar form and the use of this form in connection with operations on complex numbers. This activity illustrates a wonderful application for the angle-sum formulas and for work with polar coordinates, which students review in *Going' Round the Origin*, and gives students another perspective on complex numbers.

***Sine and Cosine Derivatives Again* (extension)** Like *Polar Complex*, this activity builds on the development of the formulas for the sine and cosine of the sum of two angles and can be assigned any time after *Comin' Round Again (and Again...)*. Students will use the angle-sum formulas to find the derivatives of the sine and cosine functions. Those derivatives were also developed in the unit *How Much? How Fast?*, using a different approach.

***Bugs in Trees* (extension)** This activity highlights an application of matrices that is quite different from the one in this unit. The activity will connect somewhat back to students' work with probability and can be used anytime after *More Memories of Matrices*.

***Half a Sine* (reinforcement)** This activity is a natural follow-up to students' work on *Doubles and Differences*.

***The General Isometry* (extension)** This activity provides a larger context for the work students are doing with translations, rotations, and reflections. It can be assigned after *Flipping Points*, which introduces reflections.

***Perspective on Geometry* (extension)** This activity is a change of pace, asking students to research the history of perspective in art, and might be used after the activity *Cube on a Screen*.

***Let the Calculator Do It!* (extension and reinforcement)** In this activity, students write a program for the calculator (or computer) to find the projection of a point on the screen given a viewpoint. This activity should follow the discussion of *Find Those Corners!*

Calculator Pictures

Intent

These activities introduce the concept of accumulation graphs and their relationship to rate graphs.

Mathematics

The first step in learning to create animation on the graphing calculator is to learn to draw pictures. Students begin by exploring the drawing commands and get some practice using a technical manual. After reviewing how to write a calculator program, they interpret a program that draws a picture and then write code for that program.

Progression

The discussion introducing *Picture This!* reveals the central unit problem: creating a program to draw an animation of a rotating cube on the calculator screen. The activity begins this task by helping students discover how to draw a picture on the screen. *Programming Without a Calculator* looks at how to create animated drawings by imbedding the drawing commands within a program.

POW 7: "A Sticky Gum Problem" Revisited introduces the first POW of the unit. Students clarify their understanding of the activity and begin work on the POW questions in *Starting Sticky Gum*.

Picture This!
POW 7: "A Sticky Gum Problem" Revisited
Starting Sticky Gum
Programming Without a Calculator

Picture This!

Intent

Students learn to draw on a graphing calculator.

Mathematics

The opening discussion introduces the central unit problem: to program an animation of a rotating cube. Students take the first step toward this in the activity, where they consult a technical manual as they explore how to draw on a calculator.

Progression

The teacher introduces the central task of the unit. Students view a calculator program that shows a rotating cube, discuss what physical motion is being illustrated, and observe a physical simulation. They then consult calculator manuals to discover how to draw on their calculators, taking notes and sharing their discoveries with others. To conclude the activity, the teacher ensures that students know how to clear the calculator screen and draw circles and line segments.

Approximate Time

65 minutes

Classroom Organization

Pairs, preceded and followed by whole-class discussion

Materials

Overhead graphing calculator (for use throughout the unit)
TURNCUBE, or a similar program, entered in the calculator
Large foam cube (or similar object) and two sticks for a classroom demonstration
Calculator manuals (1 per pair of students)
Optional: A centimeter cube and two straws for each group
Optional: A video on the making of animation features

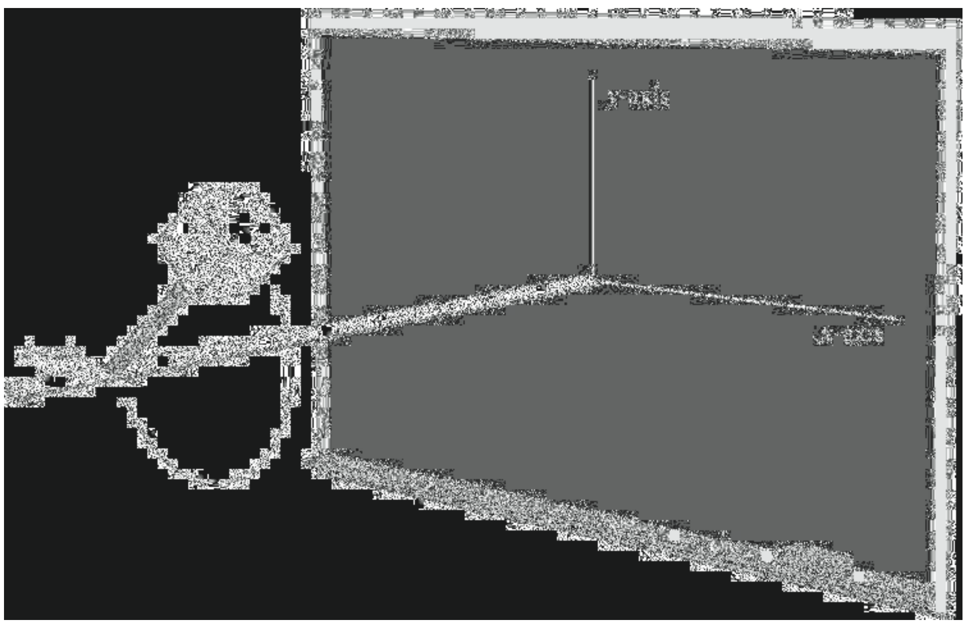
Doing the Activity

Note: If you have trigonometric identities still posted from the unit *The Diver Returns*, you may want to keep them up. If wall space is scarce, you might put them on top of one another, under another poster from this unit.

Run the program TURNCUBE (or a similar program) on your overhead calculator. The program shows a cube, situated in 3-space, making one complete rotation around the *z*-axis.

Discuss what physical motion of the cube the program is supposed to be representing, and demonstrate with a concrete model of some kind. Any sort of box, and not necessarily a perfect cube, will do. You might set this up using a classroom wall as the *xy*-plane, with the *z*-axis perpendicular to that plane. Show the cube rotating around the *z*-axis at a constant distance from the wall, keeping the same face toward the *z*-axis at all times.

As illustrated here, you might represent the *z*-axis by a stick held perpendicular to the wall and use another stick, stuck into a box (such as a foam cube), to demonstrate what's happening. The second stick will connect the box to the *z*-axis, with one end of that stick stuck into the box and the other end at and perpendicular to the *z*-axis. The second stick then turns with the end at the *z*-axis acting as a pivot point, so that the box at the other end is turned around the axis.



Bring out that as the box makes one complete revolution around the *z*-axis, it also rotates once. For instance, suppose the box starts with its "bottom" horizontal (roughly in the *xz*-plane). After the box makes a quarter turn around the *z*-axis, the face that was on the bottom will be vertical (roughly in the *yz*-plane).

You may want to give each group a cube with a hole in it and two straws and have them re-create the demonstration. They should draw the *x*- and *y*-axes on paper, hold one straw at the origin, perpendicular to the *xy*-plane, and place the second straw with one end at and perpendicular to the *z*-axis and the other end attached to the cube.

Ask, **Is the motion of the cube more like that of the moon about the earth or the earth about the sun?** If no one knows the difference, explain that the earth rotates many times as it orbits the sun once, while the moon always keeps the same face toward the earth. Thus, the cube moves the same way the moon does.

After the demonstration, run the calculator program again, now that students better understand what the program is portraying. Explain that over the course of the unit, students will learn the programming skills and mathematics concepts needed to write a program that creates the image of a moving cube on the calculator screen. You might note that although this may not seem like a very exciting animation, its program will be surprisingly complex, and students will learn techniques basic to all computer animation.

Tell students that as an end-of-unit project, they will work in pairs to create original animation programs (see *POW 9: An Animated POW*).

Optional: An Animation Video

At some point during the unit, you may want to show a video such as *The Making of Toy Story,* which describes the making of a major animation feature film. If you show such a video at the end of the unit, students will see how what they have learned is applied in real-world animation. If you show it at the beginning of the unit, students may get additional motivation.

The Plan of the Unit

Ask students, **What do you think you will need to learn to write a program to make a cube turn?** Build on their ideas to develop the following outline of the unit.

Students will learn to do these things:

1. **Draw a picture on the graphing calculator.**
2. **Create the appearance of motion.**
3. **Change the position of an object located in a two-dimensional coordinate system.**
4. **Create a two-dimensional drawing of a three-dimensional object.**
5. **Change the position of an object located in a three-dimensional coordinate system.**

You may want to post this plan for reference throughout the unit.

Tell students that not only will they learn the coordinate geometry and algebra involved in these tasks, but that they will also write programs to show these things on the calculator.

Picture This!

Have students begin work on the activity. As they work, provide opportunities for them to share discoveries with the class, perhaps using the overhead calculator. Encourage other students to question the presenters to be sure they can all replicate the results.

Emphasize that students will apply what they learn in this activity throughout the unit, so they should take careful notes. You might suggest they keep a separate notebook (or notebook section) for programming and calculator ideas and techniques.

As you circulate, ask students, **What effects would you like to create? What might you try to get the effect you want?** Encourage them to experiment. They may want to refer to the calculator manual or ask other students for suggestions.

Here are three things students should know how to do by the end of this activity:

- How to clear the calculator screen
- How to draw a line segment that connects two points given in terms of their coordinates
- How to draw a circle given in terms of its radius and the coordinates of its center

Calculator Expertise

If you do not feel completely comfortable with the programming concepts and other calculator ideas used in this unit, see the Calculator Guide and Calculator Notes.

You may also want to identify the student "calculator experts" in your class and have them assist their classmates in debugging programs and overcoming technical obstacles.

Discussing and Debriefing the Activity

Make sure student presentations cover these tasks:

- Clearing the calculator screen
- Drawing a line segment that connects two points given in terms of their coordinates (Students will need this for *Programming Without a Calculator*.)
- Drawing a circle given in terms of its radius and the coordinates of its center

Mention that because the **line** and **circle** commands involve the calculator's coordinate system, students need to adjust their viewing rectangles to see the results of these commands. Also, you may want to discuss the fact that unless the scales on the *x*- and *y*-axes are the same, circles described in terms of coordinates will look like ellipses. (Having the viewing window go from, say, –10 to 10 for both *x* and *y* will not make for equal scales unless the screen is square, which is generally not the case.)

Drawing and Programming Difficulties

Movin' On introduces the idea of creating a setup program that can be included in each new calculator program to take care of such tasks as adjusting scales and erasing previous drawings. In preparation for creating such a program, you may find it helpful to begin a poster titled "Drawing and Programming Difficulties." As students complete each activity from now until they learn about setup programs, have them add any difficulties they encounter to the poster.

An initial entry might be, "My circle is not circular." Ask the class if they ran into other problems that should be included, such as, "I forgot to erase the old picture."

Key Questions

Is the motion of the cube more like that of the moon about the earth or the earth about the sun?

What do you think you will need to learn to write a program to make a cube turn?

What effects would you like to create? What might you try to get the effect you want?

POW 7: "A Sticky Gum Problem" Revisited

Intent

In this Problem of the Week, students discover and explain patterns in a complex situation.

Mathematics

This activity is a variation on *POW 3: A Sticky Gum Problem* from the unit *The Game of Pig*. Students will look for patterns in a situation involving combinatorics and write proofs for the generalizations they find.

Progression

The activity *Starting Sticky Gum* will help students get started on this POW. Give students about a week to work on the POW, followed by presentations and discussion.

Approximate Time

10 minutes for introduction
2 to 4 hours for activity (at home)
10 to 15 minutes for presentations and discussion

Classroom Organization

Individuals, followed by several student presentations and whole-class discussion

Doing the Activity

On the day before the POW is due, select three students to make presentations on the following day.

Discussing and Debriefing the Activity

Ask the three students to make their presentations, building on what was accomplished in *Starting Sticky Gum*.

Because the emphasis in this POW is on proof, focus students' attention on finding any errors in the presenters' reasoning. List generalizations about these problems on the board or chart paper. You might make two lists—proved generalizations and conjectured generalizations—with the second being those for which no one has given a convincing proof.

Starting Sticky Gum

Intent

Students start on the new POW.

Mathematics

As students begin work on *POW 7: "A Sticky Gum Problem" Revisited* in this activity, they see a model of general formulas for such problems.

Progression

In this activity, students answer the first two questions from *POW 7: "A Sticky Gum Problem" Revisited*. In the subsequent discussion, students try to address one another's questions. Students see that the primary focus of the POW is on making and proving general conclusions.

Approximate Time

25 to 30 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

Individuals or small groups, followed by whole-class discussion

Discussing and Debriefing the Activity

Begin with a presentation on Question 1, perhaps having several students contribute. One goal of this discussion is for students to see a model of a general formula for such a problem.

Perhaps the simplest approach to this question is to examine what the maximum amount is that a parent could spend and *not* get the desired result. For instance, suppose there are n children and c colors. The worst that could happen, without getting n gumballs of the same color, is for the parent to get $n - 1$ gumballs of each color. This would cost $(n - 1)c$ cents, so the maximum that might be needed is $(n - 1)c + 1$ cents.

Go over Question 2 to be sure students see how this problem differs from the situation in Question 1. Here, the worst-case scenario is for Ms. Hernandez to get all eight yellow gumballs before getting anything else. Thus, the most she might have to spend is 9 cents.

Ask if anyone has questions about what is expected in this POW. Let other students answer the questions, clarifying as necessary.

Emphasize that write-ups for the POW should focus on general conclusions for problems like Question 2. The write-ups should also include Questions 1 and 2 (even though students have already written these up for this activity), as well as Question 3 of the POW. Students may want to revise their work from Questions 1 and 2 based on today's discussion.

Programming Without a Calculator

Intent

Students begin to learn to write a program for the calculator.

Mathematics

Students learn the distinction between a plain language-program and programming code and practice translating a plain-language program into code.

Progression

The teacher reviews the basics of writing programs for the calculator. Students then analyze a plain-language program and draw on graph paper what the calculator screen will show when the program is run. They then write code for the program, adding their own embellishments. The subsequent discussion includes time for some students to try their programs on the calculator.

Approximate Time

15 minutes for introduction
30 minutes for activity (at home or in class)
20 minutes for discussion

Classroom Organization

Individuals, preceded and followed by whole-class discussion

Doing the Activity

As needed, briefly review the general idea of writing a program for the graphing calculator.

Then illustrate (or have a volunteer illustrate) how to incorporate into a program some of the basic drawing features that students have discussed. For instance, you might demonstrate how to write and run a program to draw two line segments. Emphasize that students may need to adjust the viewing rectangle to see the results of the program.

If time allows, have students write short programs of their own to make some sort of drawing.

Programming Code and Plain-Language Programs

Mention that every programming language uses very specific syntax and commands. This formal programming language is often called *programming code,* or simply *code*. Emphasize that the code for a particular task is likely to vary from one calculator model to another. There may also be more than one way to write the code for a particular calculator.

Explain that programmers often begin with descriptions in ordinary language of what they want to do and then turn those descriptions into programming code. We will call such descriptions *plain-language programs.*

Illustrate the distinction between coded programs and plain-language programs. For instance, compare the statement "draw a line segment from (4, 2) to (7, 5)" with the programming command students need to use on their calculators to accomplish this task.

Explain that some of the activities in this unit will involve plain-language programs, including these types of tasks:

- Describing what a particular plain-language program will do when run
- "Translating" a plain-language program into code for their specific calculator
- Creating a plain-language program to accomplish a specific task

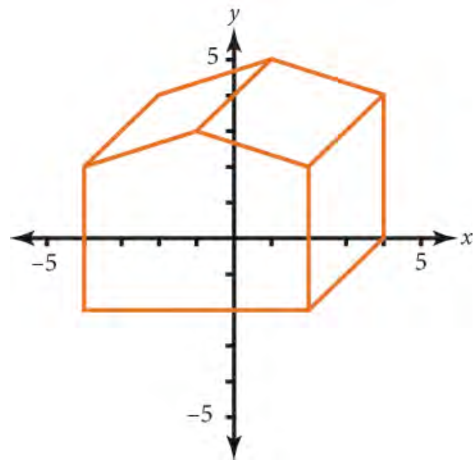
In this activity, students will perform tasks of the first two types, as well as enhance their code program to make it more interesting. Take a few minutes to go over the activity to be sure students understand what is expected in each part.

Discussing and Debriefing the Activity

Question 1

Have a volunteer show the diagram for Question 1. Meanwhile, have some students enter the code they wrote for Question 2 into their calculators and the rest enter the code they wrote for Question 3.

Here is a sample of what the diagram for Question 1 might look like. This diagram is different in some respects from what will appear on the calculator screen. If there are substantive differences of opinion about this diagram, go ahead and discuss them, but don't get sidetracked by technical details.



Question 2

Have students who entered their code for Question 2 run their programs, and compare their results to their diagrams for Question 1. They will probably see some differences, including these:

- Some of the line segments on the screen may appear jagged. You might mention that this is because of the individual pixels used to create the image.
- The screen will probably not label the axes or show values along the axes (although it may show tick marks).
- The drawings may not be centered on the screen, due to the choice of viewing window.
- The diagram may appear "stretched" vertically or horizontally, because the unit distances along the two axes are not equal.

Discuss these differences with the class. This guide will represent such calculator results using diagrams that look more like ordinary graph paper diagrams than calculator screens. You might have students use similar graph-paper-style representations (including axis labels and scales) when they are describing the output of a program. Make it clear, though, that if they are asked to write a calculator program to produce a diagram like the one just shown, they are not expected to get the calculator to label the axes, mark the scales, or create "smooth" line segments. However, they will have to choose an appropriate viewing window to make their results visible.

If students see substantial differences between the predicted and the actual results other than those just discussed, it is likely they have made errors in their code, and they should locate these errors.

Have one or two students who got a result like what they expected for Question 2 present their code and display the results. There may be variations in the code students use, so allow time for discussion.

Question 3

If time permits, have some volunteers present their enhanced programs from Question 3, both running the programs and explaining their code.

Programming Loops

Intent

In these activities, students begin to create animation programs using programming loops.

Mathematics

The programming **loop** is an essential part of writing an efficient animation program. These activities introduce the concept and syntax of “For/End” loops in graphing calculator programs. The basic steps required for animation are introduced using the analogy of a flip book.

Progression

Learning the Loops introduces programming loops. In *An Animated Shape* and *A Flip Book*, students learn how to program a simple animation and discover the usefulness of the programming loop as a means of creating a delay in a program to allow the user adequate time to view an image before it is erased.

Beginning with *Movin’ On*, students see how to use the loop variable in the drawing coordinates to greatly reduce the amount of programming code necessary for an animation. They practice this technique in *Some Back and Forth*, *Arrow*, and *Sunrise*.

Learning the Loops
An Animated Shape
A Flip Book
Movin’ On
Some Back and Forth
Arrow
Sunrise
POW 8: A Wider Windshield Wiper, Please

Learning the Loops

Intent

Students learn to use loops in programs.

Mathematics

This activity teaches students the basics of using “For/End” programming loops, including the syntax for using loops in a calculator program and the concept of **nested loops.**

Progression

The teacher introduces loops and their syntax as a shortcut for repetitive programs. Students act out what happens in the calculator as it executes a loop and then enter a loop program into their calculators. The teacher illustrates how the loop variable can be used as more than a counter. Then, given three plain-language programs using loops, students describe or sketch what will happen when they run the programs. Finally, students write their own programming code utilizing a loop.

Approximate Time

30 minutes for introduction
30 minutes for activity (at home or in class)
20 minutes for discussion

Classroom Organization

Individuals or groups, preceded and followed by whole-class discussion

Materials

Optional: Transparency of *Learning the Loops* blackline master

Doing the Activity

Begin with a simple example to motivate the use of programming loops. Ask students to work in their groups to write a calculator program that will generate a screen display like this one:

```
HI
BYE
HI
BYE
HI
BYE
```

Students may have already discussed (for *Picture This!*) the instructions for getting the calculator to display text on its screen. If not, introduce the necessary code.

Students will probably create a program that contains a separate instruction for each line of the desired display. In plain-language form, the program might look like this:

> Display the word "HI" on the screen
>
> Display the word "BYE" on the screen
>
> Display the word "HI" on the screen
>
> Display the word "BYE" on the screen
>
> Display the word "HI" on the screen
>
> Display the word "BYE" on the screen

This plain-language program ignores such details as the exact position of each display. You may want to mention these details in the discussion. Often, each new display instruction will place the desired text at the beginning of a new line, which is the goal here.

The Plain-Language For/End Loop

Bring out the fact that the same pair of instructions is being repeated three times, and ask, **How might you avoid this repetition in the program?** Explain that most programming languages have a mechanism (and perhaps several mechanisms) for performing the same set of instructions more than once.

If you can use students' ideas to develop this loop, fine. Otherwise, have a student give you a letter to use as a variable. Then show the class this plain-language program:

> For A from 1 to 3
>
> Display the word "HI" on the screen
>
> Display the word "BYE" on the screen
>
> End the A loop

Explain that this program uses the instructions "For" and "End" to create a programming device called a **loop.** Also introduce the related terminology. In this example, the letter A is called the *loop variable*. The numbers 1 and 3 are called, respectively, the *initial value* and the *final value* for the loop variable. The two lines between the "For" and "End" instructions form the *body* of the loop.

Tell students that to show the overall structure of the plain-language program more clearly, we sometimes indent the body of the loop, with a bullet for each step. Thus the program might be written like this:

For A from 1 to 3

- Display the word "HI" on the screen
- Display the word "BYE" on the screen

End the A loop

Acting Out the Program

Ask students, **How do you think this program works?** Their answers may be somewhat speculative, but some students will likely be able to describe the general idea.

After some verbal descriptions, have the class act out the program, perhaps using a box drawn on the board (or an actual box) as the *memory cell* for the variable A. (Students should be familiar with the idea of a memory cell for a variable from earlier work with programming.) Also set up something to serve as the calculator screen.

Although calculators may vary slightly in how they actually carry out a loop like this, you might have students act it out using this scenario:

- They start with the "For" instruction, and interpret that by putting the number 1 in the box labeled A.
- They carry out the body of the loop by printing "HI" and then "BYE" on the "screen."
- For the "End" instruction, they increase the value of the variable A (the number in box A) to the next value (which at this stage means going from 1 to 2) and then go back to the "For" instruction.
- When they return to the "For" instruction, they check whether the loop variable exceeds 3. It does not (because A is now 2), so they proceed.
- They carry out the body of the loop, printing "HI" and then "BYE" again.
- When they reach the "End" instruction, they increase the value of A from 2 to 3, and again go back to the "For" instruction.
- When they return to the "For" instruction, they check whether the loop variable exceeds 3. It does not (because A is now 3), so they proceed.
- They carry out the body of the loop, printing "HI" and then "BYE" (for the third time).
- When they reach the "End" instruction, they increase the value of A from 3 to 4, and again go back to the "For" instruction.
- When they return to the "For" instruction, they check whether the loop variable exceeds 3. This time it does (because A is now 4), so the loop is completed.

Explain that if there were any instructions after the "End" instruction, students would proceed to the first such instruction. In this example, there are no such instructions, so the program is over.

Writing the Program in Code

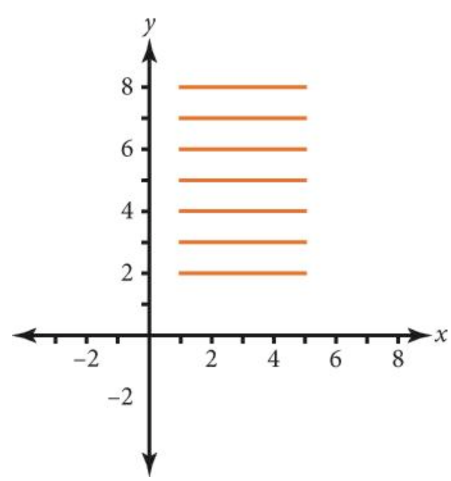
Go over how students can write this simple loop in code for their calculators. There may be more than one option, but many calculators have a loop format that uses the specific instructions "For" and "End"; have students use this type of loop if possible. Ask students to enter and run the program on their calculators to verify that it works.

Note: In some calculators, certain letters serve special purposes, such as using *x* and *y* for defining functions. Warn students never to use such letters as variables in their programs. Also, it's easy to accidentally write a program that sets up an infinite loop. There should be a key on the calculator to interrupt a program while it is running. You may want to introduce that key to students.

Using the For/End Variable

Ask students, **What role does the variable A play in this program?** They should see that it keeps track of how many times the pair of instructions to display "HI" and "BYE" is carried out. In such an example, the loop variable is sometimes called a *counter.*

The next step in understanding the use of the For/End loop is to see how the loop variable can be used in a way beyond simple counting. Show the class the accompanying diagram (also supplied as a blackline master).



Have students work in groups to write a program to produce these line segments on the calculator screen. They don't need to get the calculator to label the axes or show numeric scales, but they do need to set an appropriate viewing window.

Some students may recognize that they can use a loop to do this, but others will probably come up with something analogous to this plain-language program:

- Draw a line segment from (1, 2) to (5, 2)
- Draw a line segment from (1, 3) to (5, 3)
- Draw a line segment from (1, 4) to (5, 4)
- Draw a line segment from (1, 5) to (5, 5)
- Draw a line segment from (1, 6) to (5, 6)
- Draw a line segment from (1, 7) to (5, 7)
- Draw a line segment from (1, 8) to (5, 8)

If no one suggests using a loop, ask, **How might you use a loop to shorten the program?** If necessary, point out that these instructions are all quite similar or ask what is changing in the program. Someone should be able to come up with a program like this:

For B from 2 to 8
- Draw a line segment from (1, B) to (5, B)

End the B loop

You may want to have students act out how this works. At least go over the fact that the initial value is 2, so the calculator begins by putting the number 2 in a memory cell labeled B.

If students are confused by the fact that the loop variable is starting at a value other than 1, point out that the *y*-coordinate is changing from one line of the original program to the next, and ask, **What is the *y*-coordinate for the first line segment that needs to be drawn?**

Note: Students who want to use 1 for the initial value can do so by having the body line of the loop be

- Draw a line segment from (1, B+1) to (5, B+1)

In this case, the "For" instruction would read

For B from 1 to 7

Entering and Running the Program

Have students enter and run the new program and check that it works properly. As with all drawing programs based on coordinates, students may need to adjust the viewing rectangle to get the desired picture. This is a good occasion to look at how to set the viewing rectangle within the program itself. This discussion can be delayed and included in connection with the "setup" program discussed following *Movin' On*, but in any case add this issue to the class poster of programming difficulties encountered.

Learning the Loops

Students are now ready for the activity, which will give them practice using loops in programs.

Discussing and Debriefing the Activity

Question 1

Question 1 should be straightforward. Make sure students see that the loop variable is acting simply as a counter.

Question 2

Have students act out this program. They will need two boxes, one for each variable. The program should create an output sequence like this:

LEMON
LIME
LIME
LIME
LEMON
LIME
LIME
LIME
LEMON
LIME
LIME
LIME
LEMON
LIME
LIME
LIME
LEMON
LIME
LIME
LIME

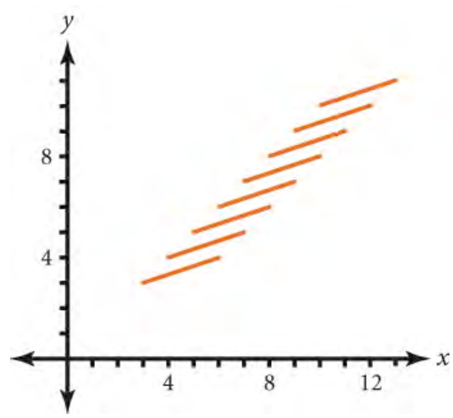
Introduce the term **nested loop,** which means a loop within a loop, for the programming structure involved in Question 2.

Ask students, **How would you write the code for this program?** Point out that in our plain-language programs, the "End" instruction states what loop it refers to, but in most actual programming code, it does not. The general convention is that an "End" instruction sends the program back to the most recent "For" instruction.

Question 3

Ask for volunteers to describe what they think will happen in Question 3. It's important that students feel comfortable making such predictions. Assure them that even experienced programmers often make mistakes in interpreting instructions.

Then have students turn this plain-language program into code and try out their programs to confirm their predictions. Of course, the screen display will depend on the viewing rectangle, and the line segments will probably appear rather jagged. However, the result should look something like this:



Question 4

Have students enter the programs they created into their calculators and perhaps share their work in their groups. You might have some students show their programs to the class using a projector.

Key Questions

How might you avoid this repetition in the program?
How do you think this program works?
What role does the variable A play in this program?
How might you use a loop to shorten the program?
What is the *y*-coordinate for the first line segment that needs to be drawn?
How would you write the code for this program?

Supplemental Activity

***Loopy Arithmetic* (extension)** challenges students to perform more difficult programming tasks using loops.

An Animated Shape

Intent

Students make their first animation program.

Mathematics

Using the analogy of a flip book as an illustration of how an appearance of movement can be created with stationary pictures, students learn to create animation on the calculator.

Progression

The teacher introduces this activity with a discussion of where the class is in the unit and uses a flip book to introduce the topic of animation. Students then create a calculator program that works much like a flip book, drawing a picture in one location and then repeatedly and quickly erasing it and displaying it in a new location. They will likely encounter the difficulty that their pictures are erased too quickly to even view them, which leads into a discussion of how to create a *delay loop.*

Over the next few activities, students will look at how to create the illusion of motion. They will then consider how to change the position of two-dimensional objects (*Translation in Two Dimensions* and *Rotation in Two Dimensions*), how to create a two-dimensional drawing of a three-dimensional object (*Projecting Pictures*), and how to change the position of three-dimensional objects (*Rotation in Three Dimensions*).

Approximate Time

70 minutes

Classroom Organization

Pairs, preceded and followed by whole-class discussion

Materials

Optional: A sample flip book (perhaps one made by a former student)

Doing the Activity

Remind students that their goal is to create an animated picture of a cube turning. Summarize that they now know something about drawing on the calculator and are learning about the calculator's programming language so they can write appropriate instructions.

Refer to the plan posted during the discussion introducing *Picture This!* Students can now check off item 1: "Draw a picture on the graphing calculator." Then review the remaining items, which you might organize into three categories:

- Creating the illusion of motion on the calculator screen
- Changing the position of an object located in terms of a coordinate system
- Creating a two-dimensional drawing of a three-dimensional object

This is a good time to review the terms *2-space* and *3-space* to refer to the sets of all points in the two-dimensional and three-dimensional coordinate systems.

Flip Books: The Illusion of Motion

Ask if anyone can explain the basic idea of animation. **What is animation? How does it work? How could you create it?** Highlight these two key ideas:

- Animation involves a series of drawings, each differing slightly from the previous one.
- The illusion of motion is created by showing each picture very briefly and then replacing it by the next.

Mention that one simple device for showing animation is a flip book. If possible, illustrate with an example. Otherwise, describe how a flip book can be made with a set of index cards, illustrating how to hold and flip through the cards so that each shows for only a fraction of a second. Tell students they will soon create their own flip books using a set of index cards (in *A Flip Book*), as well as applying the principle in today's activity.

An Animated Shape

This is a good activity for students to work on in pairs. Stress that students should not get so carried away with artistry on their initial picture (Question 1) that they don't have time to complete the remaining questions.

Students will probably realize they need to erase the screen before drawing each new picture. But they may not realize that they need to delay before doing so to allow time for the picture to register on the brain. The mechanics of introducing a delay into a program should come up in the discussion of this activity. If you want to plant the seed of that idea, ask pairs to think about why they see only one picture (the last one) when they run their programs. (We use the term *delay* rather than *pause* because in some calculators a "Pause" command simply halts the program until the user tells the calculator to resume.)

Students who recognize the need for a delay may become frustrated if they are unable to get the calculator to do what they want. If they can't come up with a solution, assure them that this will be covered in the discussion of the activity.

Discussing and Debriefing the Activity

Begin by asking students to describe their results, perhaps having one or two pairs demonstrate their programs on the calculator overhead projector. Ask, **What worked and what didn't work?** Have students share any mistakes they made and any difficulties they encountered.

As part of the discussion, ask, **What steps were involved in creating the illusion of motion?** Three key issues need to be identified:

- They need to erase their first picture before drawing the second.
- They need to know how to show the same picture in a different position.
- They need to delay before erasing each successive drawing. Otherwise, all drawings but the last will blur by very quickly, and they will only really see the last drawing.

Most likely, students will have figured out that they need to include an instruction in their programs to erase the previous drawing. But they may have some trouble with the details of the other two issues, especially the question of how to make the calculator delay before erasing.

The details about how to move an object are the essence of items 3 and 5 in the unit outline, from *Picture This!,* and students will begin work on these details in *Move That Line!*

The Delay

Ask for suggestions about how to create a delay in a program. **How can we get the calculator to delay before it clears the screen?**

One solution is to have the calculator run through a loop with no body instructions. To suggest this idea, you can propose that students give the calculator something to do to keep it busy before it clears the screen. One such task would be to have it count to some number, such as 200. Here is the basic structure of such a program:

```
For T from 1 to 200
End the T loop
```

We will refer to this as a *delay loop.* Make sure students recognize that they can control the duration of the delay by adjusting the final value of the loop variable.

Key Questions

What is animation? How does it work? How could you create it?
What worked and what didn't work?
What steps were involved in creating the illusion of motion?
How can we get the calculator to delay before it clears the screen?

A Flip Book

Intent

Students create flip books to further understand the idea of animation.

Mathematics

The discussion introducing *An Animated Shape* used the analogy of a flip book to illustrate how animation is created by revealing an image and then quickly replacing it with a series of images of similar objects in slightly different positions. This activity reinforces that idea to help students understand what they will be trying to accomplish when programming an animation.

Progression

Students create a simple flip book and share their work in groups.

Approximate Time

30 minutes for activity (at home or in class)
10 minutes for sharing flip books

Classroom Organization

Individuals, followed by groups

Materials

Index cards (6 to 8 per student)

Doing the Activity

The concept of a flip book was introduced in the discussion preceding *An Animated Shape*. Give students index cards to use in making their flip books.

Discussing and Debriefing the Activity

Rather than discuss the activity, you might simply give students a few minutes to share their flip books in their groups.

Project Reminder

While students are sharing their flip books, remind them that this unit includes a final project in which they will work with a partner to create an animation program on the calculator (*POW 9: An Animated POW*). Point out that although they may not know enough yet to begin writing the program, they can begin thinking about a subject for their project and considering who to work with.

Movin' On

Intent

Students analyze what a specific program involving animation will do.

Mathematics

You can use this activity to assess students' understanding of loops. With students having created their first animation program in *An Animated Shape*, this gives them another model of how a loop can be used in a program to create the effect of animation.

This activity also introduces the concept of a *setup program,* a program that can be used and easily modified to accomplish such initial tasks as clearing the screen and adjusting the viewing window.

Progression

Students describe the result of a plain-language program that uses a loop to create animation and then create the programming code for the program. They also make a list of components that might be included in a setup program. In the subsequent discussion, students test their programming code and share ideas about what to include in a setup program.

Approximate Time

30 minutes for activity (at home or in class)
50 minutes for discussion

Classroom Organization

Individuals, followed by whole-class discussion

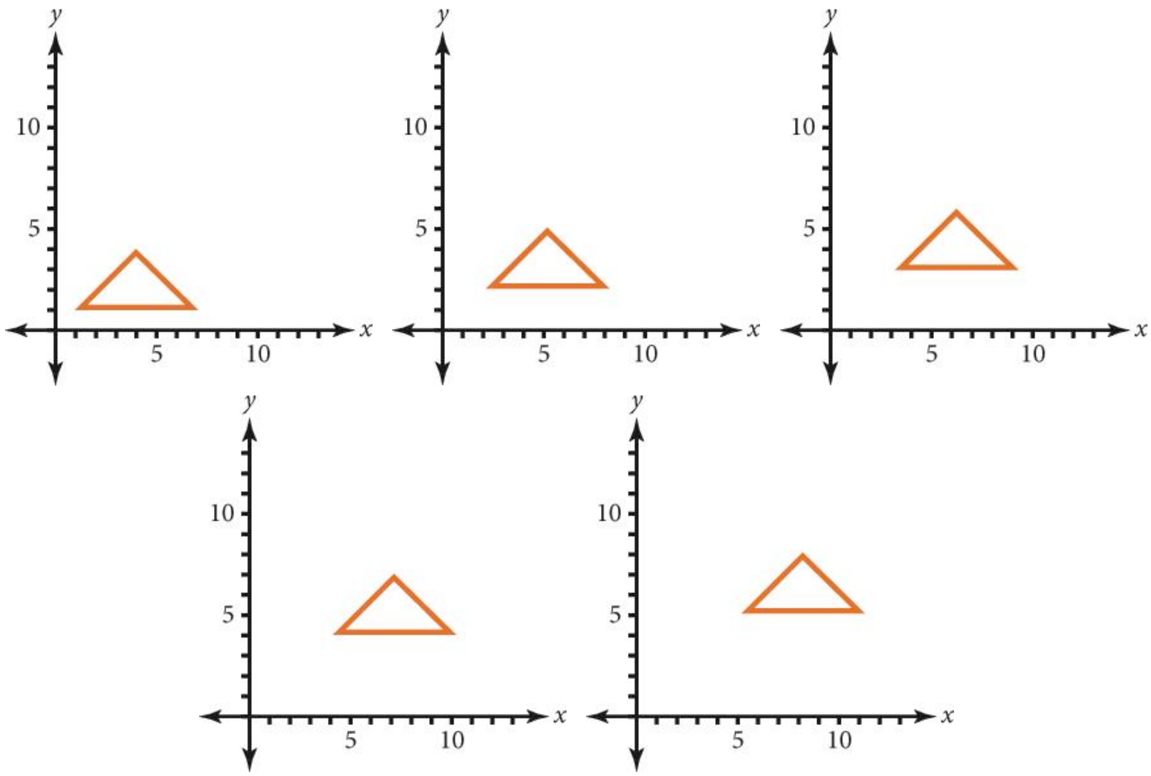
Doing the Activity

This activity requires no introduction.

Discussing and Debriefing the Activity

Question 1

Have a volunteer present part a, which should involve a sequence of diagrams like this:



Give students time to enter the code they created for part b to see if it gives the desired result. Then have one or two students present successful code (or discuss this as a class if no one was able to create the appropriate code).

Question 2

Begin the discussion of this question by asking, **Why might a setup program be needed or desirable?**

Students have probably seen that when they run programs involving a coordinate display, they often need to adjust the viewing window. Some may also have run into the dilemma of having functions graphed over their work or of having to clear an old graphic from the screen. A setup program can be used to eliminate these stumbling blocks. Here are two ways students can incorporate such a program into their work:

- They can write a separate program called "Setup" and have "Run the setup program" be the first instruction of each drawing program they write.

- They can copy the individual instructions of the setup program into the beginning of each drawing program they write.

Ask, **What instructions should we include in a setup program?** If you have been keeping a class poster of programming difficulties encountered, the class may find several ideas there. Here are some possibilities:

- Clear the screen.
- Make the function graphing feature inactive.
- Set the viewing rectangle (the parameters in this part of the setup program will need to be adjusted to suit the individual program).

Have students develop code for each of these components and test it on a program. They should also keep a written record of these ideas.

Students will no doubt encounter other items to include in a setup program as they progress through the unit. For instance, when they start working with angles, they may want to have an instruction that sets the "mode" for angles to degree measurement.

Key Questions

Why might a setup program be needed or desirable?
What instructions should we include in a setup program?

Some Back and Forth

Intent

Students practice creating animation with a program using a loop.

Mathematics

This activity continues the ideas of *Movin' On*, with students now writing the plain-language animation program as well as the programming code for an animation, including a description of their setup program.

Progression

Students write a program using a loop to create eight drawings of a line segment moving back and forth between two positions. They provide both a plain-language program and programming code, including a setup program.

Approximate Time

30 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

Individuals, followed by whole-class discussion

Doing the Activity

This activity requires no introduction.

Discussing and Debriefing the Activity

Begin with a discussion of the plain-language program, and have one or two students present their ideas. Here is one option:

Setup program
For A from 1 to 4
- Clear the screen
- Draw a line segment from (1, 1) to (2, 2)
- Delay
- Clear the screen
- Draw a line segment from (2, 2) to (3, 3)
- Delay

End the A loop

Some students may use nested loops, saving one instruction line. If so, their programs are worth sharing.

Have students enter their programming code into the calculators to see if it creates the desired sequence of line segments. This is a good opportunity for them to try out their ideas about setup programs.

Arrow

Intent

Students continue to practice using a programming loop.

Mathematics

Students create a simple animation using a programming loop.

Progression

Given a plain-language program for an animation of a flying arrow, students create programming code, test it in their calculators, and then add their own embellishments.

Approximate Time

20 to 25 minutes

Classroom Organization

Individuals

Doing the Activity

No introduction to this activity is necessary.

Discussing and Debriefing the Activity

This activity probably needs no discussion unless students want to share their embellished programs.

Sunrise

Intent

This activity provides further practice with using a program to create animation.

Mathematics

This activity gives students a chance to apply their new ideas about animation, as well as to use their calculator's capacity for drawing circles.

Progression

Students create a program to display an animation of a rising circular "sun" by sketching the displays on graph paper, writing a plain-language program, and writing programming code. The subsequent discussion is a nice opportunity to review the use of setup programs.

Approximate Time

5 minutes for introduction
30 minutes for activity (at home or in class)
20 minutes for discussion

Classroom Organization

Individuals, followed by whole-class discussion

Doing the Activity

Take a few minutes to remind students of the programming code for drawing circles on their calculators. The code will probably use three parameters: two for the coordinates of the center and one for the radius.

Discussing and Debriefing the Activity

Have students test their programs by entering them into their calculators. While they are working, have one or two students put their plain-language programs on a transparency to share. Also have a couple of students use the overhead calculator to display their programs to the entire class.

If it hasn't yet come up, bring out that unless the scales on the x- and y-axes are the same, the circles will look like ellipses.

The issue of scales provides another context for discussing the use of a setup program (see the discussion notes for *Movin' On*). Focus specifically on the issue of setting the viewing window, which will vary from program to program. In the case of *Sunrise,* the viewing window has to be set to make the circles visible (or parts of them, perhaps), as well as to make the scales on the x- and y-axes the same.

Where Are We in the Unit?

Once again, have students look at the outline of the unit. They should see that they can check off item 2: "Create the appearance of motion."

POW 8: A Wider Windshield Wiper, Please

Intent

Students solve and explain a complex problem that involves maximizing area.

Mathematics

This POW involves finding the areas of irregularly shaped regions partially bounded by curves. Students must also find the area of part of a circle, find the length of a hypotenuse, and recognize that a particular curve is not a circular arc.

Progression

Students are first challenged to determine which of two designs using a particular windshield wiper will wipe the largest area and then to come up with a design of their own that will wipe a greater area. They are reminded that this POW does not use the standard format for the write-up. The activity concludes with student presentations.

Approximate Time

5 minutes for introduction
2 to 3 hours for activity (at home)
15 to 20 minutes for presentations

Classroom Organization

Individuals, followed by presentations and discussion

Doing the Activity

Make sure students realize that their reports for this POW should be similar to a business analysis, not the standard POW write-up. You may want to set some guidelines about the length of these reports. You might also suggest that students examine the windshield wipers on some actual cars to see how they function.

Give students about a week to work on this POW.

On the day before the POW is due, choose three students to make presentations on the following day.

Discussing and Debriefing the Activity

Ask the selected students to make their presentations. The two wipers don't clean the same area, and it will be interesting to see whether students can find a wiper that covers more area than either model.

A Flawed Explanation

You may find it interesting to examine the following argument, which appears to show that the two wipers clean the same area.

> The standard wiper cleans one-fourth of the area between the two concentric circles with radii of 18 inches and 6 inches. So that wiper cleans an area of 72π square inches.
>
> The second wiper is also a blade with length 12 inches. As it moves from left to right, its midpoint moves through a path that is one-fourth of the circumference of a circle with radius 12 inches. (This is the path of the tip of the arm.) This path has length 6π inches. So the blade sweeps a 12-inch vertical strip across a path of length 6π inches, for an area of 72π.

It turns out that part of the argument regarding the second wiper is incorrect. What exactly is wrong with this argument, besides the fact that you get the wrong answer? (*Hint:* Think about comparing the area of a rectangle with the area of a parallelogram. Some ideas from *The Leading Edge* in the unit *How Much? How Fast?* may lend further insight. The area covered by the second wiper is actually $144\sqrt{2}$ square inches.)

Translation in Two Dimensions

Intent

In these activities, students learn to use matrices to perform geometric translations.

Mathematics

To produce smooth motion at a constant speed, subsequent pictures in an animation sequence need to represent **translation** by a constant vector. The activities in this section introduce this idea to students as a *translation vector* and use this vector in matrix addition to perform translation in an animation program.

Progression

Move That Line! introduces the concept of a translation vector, and *Memories of Matrices* incorporates the use of that vector with matrix addition to accomplish animation. *Double Dotting* gives students practice with a skill that is critical for this application of matrices: using variables in loops as geometric coordinates.

In preparation for the development of trigonometric identities needed for performing geometric rotations with matrices, *Cornering the Cabbage* develops the general formula that the area of a triangle is equal to the product of two sides times the sine of the included angle.

Move That Line!
Double Dotting
Memories of Matrices
Cornering the Cabbage

Move That Line!

Intent

Students learn how to translate graphics.

Mathematics

This activity introduces geometric transformations and the use of a translation vector to accomplish a translation. Students use delays and loops to move a segment repeatedly by the same translation.

A transformation that maintains the shapes of figures is called an **isometry.** The term, which is introduced to students in *Flipping Points,* comes from Greek roots meaning "same measure" and refers to the fact that the distance between the images of two points is the same as the distance between the points themselves. Isometries are also called *distance-preserving functions* or *rigid motions*. Students will be learning about the three basic types of isometries: translations, rotations, and reflections. Translations are introduced in this activity, rotations in *Goin' Round the Origin*, and reflections in *Flipping Points*. In *Another Mystery,* students will combine a translation with a rotation, and they may see that the result is actually another rotation.

Progression

The teacher introduces this activity by defining **geometric transformations** and **translations** and developing the representation of a translation by a translation vector. Students draw a line segment on graph paper, choose a translation, and repeatedly apply the translation to the segment. They then write and test a program to produce the same drawing. Students who accomplished this without using programming loops modify their programs to use them. Finally, students use the same procedure to make a program that moves a simple picture composed of several line segments.

The subsequent discussion focuses on the use of a loop to repeat a translation and introduces a generic outline for an animation program.

Approximate Time

70 minutes

Classroom Organization

Small groups or individuals, preceded by direct instruction and followed by whole-class discussion

Doing the Activity

Ask students to turn their attention to the class outline for this unit. They will now begin work on the task described in item 3:

> **3. Change the position of an object located in a two-dimensional coordinate system.**

Tell students that the method for doing this will involve a point-by-point function that says, in effect, where each point ends up when it is moved. Explain that mathematicians often use the word *transformation* for geometric functions, but that students will be looking at a very special type of function, because they don't want to change the shape of objects when they change the positions of individual points.

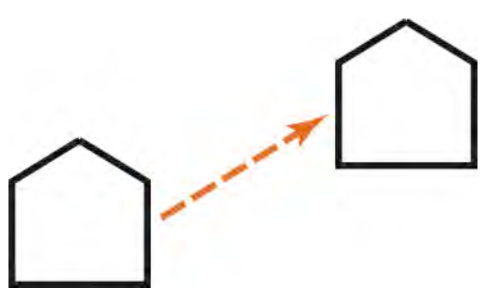
You may want to focus on a figure such as a triangle and ask, **What should be true about the new figure compared to the old one?** Bring out that the new figure should be *congruent* to the old one.

Have students picture a triangle as a physical object, such as might be made out of cardboard, and ask, **In what ways can this triangle be moved without changing its shape?** Try to elicit a variety of descriptions. You might get phrases like "move it sideways or up or down," "turn it," and "flip it over." Tell students that each of these motions is a possibility and that they will look at each case over the course of the unit.

Translations

Tell students that one of the basic transformations they will study is a **translation,** also known informally as a *slide.* Describe a translation as a transformation in which every point is moved the same distance in the same direction.

Draw a simple diagram, like the outline of a house, and have a volunteer draw what he or she thinks the result of *translating* or *sliding* this diagram might be. You should get a diagram something like this:

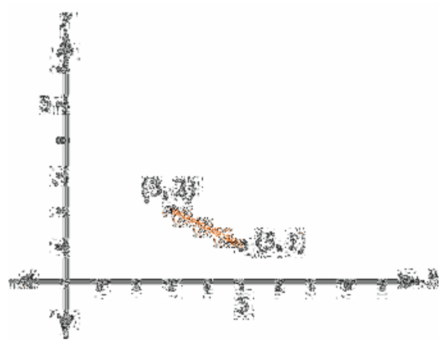


Explain that the term *translation* is used to refer both to the *process* of moving the points and to the *result* of the movement. We might say that we have translated the first house up and to the right and that the second house is a translation of the first.

Numeric Description of a Translation

Tell students that in most of their work on the central unit problem, they will be giving instructions to the calculator in terms of coordinates. Therefore, it will be important for them to be able to describe translations numerically.

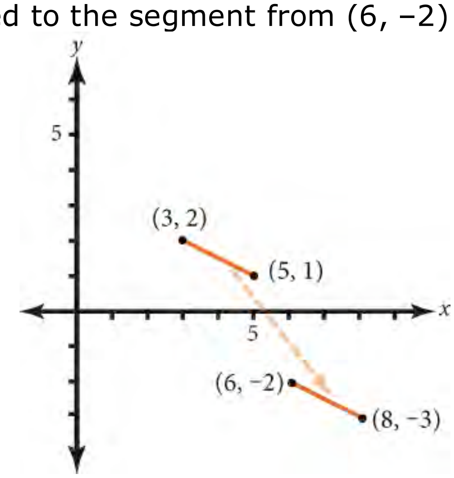
As an example, have them look at the line segment from (3, 2) to (5, 1). You may want to show this segment on a coordinate grid.



Ask each group to find examples of segments that are translations of this initial segment. Because a segment is defined on the calculator by its endpoints, students have to decide what new pairs of endpoints would work.

Then ask, **What could you do to the original coordinates to get the translated segment?** The goal is for students to see that they can get the endpoints of a translated segment by adding a particular number to the two *x*-coordinates and adding a particular number, possibly different from the first, to the two *y*-coordinates.

For instance, if they add 3 to both *x*-coordinates and −4 to both *y*-coordinates, the initial segment is translated to the segment from (6, −2) to (8, −3).



Tell students that this translation can be represented by (3, −4), and ask them to explain this representation. **Why can this translation be represented by the ordered pair (3, −4)?** They should see that the first number represents how far each point will move in the horizontal direction (to the right or left) and the second represents how far each point will move in the vertical direction (up or down). You may want to talk about the role of the sign of each number.

Tell students that in this context, the ordered pair (3, −4) is called a *translation vector* and is written [3 −4]. Identifying this vector is the first step toward the use of matrices in representing translations. *Memories of Matrices* will review basic ideas about matrices. For now, you may simply want to remind students that a vector is a matrix with only one row or only one column.

Ask what this translation would do to some general point. **What does this translation do to the point (*a, b*)?** Students should see that it would move (*a, b*) to (a + 3, b − 4). Bring out that this corresponds to the sum of the two row vectors [*a b*] and [3 −4].

Point out the analogy between representing the translation by the two numbers 3 and −4 and the idea of *components of velocity* that students used in *The Diver Returns*. In both cases, motion in two dimensions is expressed as a combination of vertical and horizontal parts.

The approach just described may not be the way students actually found the translated segments. For instance, they may have seen that one of the original endpoints, (5, 1), was "two over, one down" from the other original endpoint, (3, 2). They might then have picked one new endpoint arbitrarily and found the other new endpoint by going "two over, one down." If students describe this alternate method, acknowledge that it is correct. But bring out that the "two over, one down" approach does not provide an effective description of the translation overall, because it does not explain what should happen to *every* point.

Move That Line!

Tell students that their next programming challenge is to write a program that moves a line segment repeatedly across the screen, using the same translation for each movement. With that introduction, have them begin work.

If groups have trouble getting started, tell them to pick endpoints for their initial segment and decide where the segment should move to in the first translation. This is essentially what they are asked to do in Questions 1a and 1b.

At least some groups likely will initially write programs for Question 2 without using a loop, and this discussion is based on that assumption. A plain-language version is given in the next section below.

Groups may need help with Question 3. In particular, you may need to clarify the process by which students find the coordinates: **How do you find the coordinates of the endpoints of each new segment from those for the previous segment?** You may also need to give groups assistance setting up the variables.

For example, they might call one endpoint (P, Q) and the other (R, S). If they start with the segment from (3, 2) to (5, 1), they will need steps in their program that set P to 3, Q to 2, R to 5, and S to 1. They can then move the segment by changing the values of the variables P, Q, R, and S. A sample plain-language program is included in the follow-up discussion of this activity.

Do not expect all groups to get to this stage quickly. Some may only get through Question 3. Question 4 is included for faster groups.

You may want to remind students to begin their programs with the setup program, so the screen is cleared at the beginning and the viewing window is set to appropriate values.

Discussing and Debriefing the Activity

This discussion assumes that at least some groups did Question 2 without the use of loops. "Question 2" will mean a program written without loops, and "Question 3" will mean an approach using loops. If all groups used loops initially when they worked on Question 2, go straight to Question 3 when you begin the discussion.

Let students continue work until most have finished Question 3. As some groups complete Question 3, ask a member of one group to prepare a presentation on Question 3. Also, if possible, identify a group that did Question 2 initially without loops and have a member of that group prepare a presentation.

Question 2: Moving the Line Without a Loop

Have the student presenting Question 2 go first, to validate this more concrete approach and to lay a foundation for the use of a loop for repetition of a translation.

A plain-language program for part a might look like this. The specific numbers will depend on the initial segment and the translation that groups choose. We are also assuming that the setup program includes a "clear screen" instruction.

```
Setup program
Draw a line segment from (3, 2) to (5, 1)
Delay
Clear the screen
Draw a line segment from (6, –2) to (8, –3)
Delay
Clear the screen
Draw a line segment from (9, –6) to (11, –7)
```

Delay
Clear the screen
Draw a line segment from (12, –10) to (14, –11)
Delay
Clear the screen
Draw a line segment from (15, –14) to (17, –15)
Delay
Clear the screen
Draw a line segment from (18, –18) to (20, –19)

Post the plain-language program for reference during the rest of the discussion.

Question 3: Moving the Line with a Loop

Now have a presentation on Question 3. Here are the key elements for writing the program using a loop:

- Using variables to represent the coordinates of the endpoints
- Writing instructions by which these coordinates are changed

As needed, use the more concrete approach of Question 2 to help students understand the use of the loop. For instance, ask how the numbers in the last instruction for Question 2a were obtained from the numbers in the previous instruction for drawing a line segment. **How do you get the pairs (18, –18) and (20, –19) from the previous pairs?**

Emphasize that the same thing is done to get each of the new *x*-coordinates from the previous *x*-coordinates. For instance, in the example given, students would add 3 to both 15 and 17 to get the new *x*-coordinates 18 and 20. The same thing is done to get each new *y*-coordinate from the previous *y*-coordinate. In our example, we are adding –4 (or subtracting 4) in each case. (In anticipation of the discussion of translation matrices, it may be helpful to suggest that students think of the arithmetic as adding –4 rather than subtracting 4.)

Help students see that the pair of numbers used to get the new coordinates (in this case, 3 and –4) corresponds to the translation vector [3 –4] used in the previous discussion ("Numeric Description of a Translation") to describe the translation.

The identification of this translation vector may be helpful for some groups in making the transition to the use of a loop for repeated translation, as it makes explicit the process of obtaining the new coordinates from the old ones.

Emphasize that the same process is used at each stage to get the next pair of points. Focus on the fact that the repetitious nature of the task makes it suitable for a loop.

Post for later use at least one of the programs developed that uses a loop. Here is one possible plain-language program:

Setup program
Set P equal to 3, Q equal to 2, R equal to 5, and S equal to 1
For N from 1 to 6
- Clear the screen
- Draw the line segment from (P, Q) to (R, S)
- Delay
- Add 3 to each of P and R, and add –4 to each of Q and S

End the N loop

The specific numbers used will vary from group to group, as might the structure of the programs.

The Generic Animation Outline

Bring out that a loop can be used for many animation programs. At least three things will vary from one animation program to another:

- How the initial coordinates are set
- What gets drawn using the coordinates
- How the coordinates are changed

As long as there is some way to describe how to get the next set of coordinates from the previous set, this approach will work. For translation, the coordinate change consists of adding the same number to each x-coordinate and the same number to each y-coordinate.

Post an outline like this one (or something similar developed by the class) as a "generic" animation outline:

Setup program
Set the initial coordinates
Start the loop
- Clear the screen
- Draw the next figure
- Delay
- Change the coordinates

End the loop

Note: When this program goes through the loop for the first time, the "next" figure is actually the "first" figure. Alternatively, the program could be written so the first figure is drawn before the loop begins. Also, because the loop in this outline includes an instruction to clear the screen, the setup program need not include one.

Point out that this outline is much more of an overview than the plain-language programs students have been writing. Tell them they will create outlines as part of their work on their animation projects. This will involve an overview like the one just developed, but more specific to their particular projects.

Question 4

Use Question 4 as a vehicle for pointing out that the process works even for more complicated diagrams. There would be more pairs of variables, but one would simply assign them initial values and change them the same way as was done with the four variables used previously.

If time permits, have students share their results from Question 4 on the calculator overhead projector.

Key Questions

In what ways can this triangle be moved without changing its shape?
What could you do to the original coordinates to get the translated segment?
Why can this translation be represented by the ordered pair (3, −4)?
What does this translation do to the point (*a*, *b*)?
How do you find the coordinates of the endpoints of each new segment from those for the previous segment?
How do you get the pairs (18, −18) and (20, −19) from the previous pairs?

Double Dotting

Intent

Students use a loop to create a program with a geometric translation.

Mathematics

This activity will help students get a handle on using variables as geometric coordinates.

Progression

Given two plain-language programs that use variables in loops to draw a pattern of dots, students sketch what should appear on the screen and then write a similar program of their own.

Approximate Time

25 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

Individuals, then groups, followed by whole-class discussion

Doing the Activity

Students work on this activity independently.

Discussing and Debriefing the Activity

Have students share the programs they wrote for Question 3 in their groups. While this is happening, choose two students to prepare presentations of Questions 1 and 2. Ask them, as part of their presentations, to go step by step through the program, showing how the values of the variables change at each step so other students can see what is happening. For instance, they might use boxes to show the value of each variable or make a table with a column for each variable. Then ask them to give their presentations.

Memories of Matrices

Intent

Students review matrices and see how to use them for translations.

Mathematics

This activity will help students recall what they learned about matrices in Year 3. During the discussion, they will begin using matrices to work with drawings. This notation will be especially helpful when they encounter rotations.

Progression

Students refamiliarize themselves with how to add matrices and are then led through the steps of using matrices to represent and translate the coordinates of the segment endpoints in a simple drawing. The activity also reviews the notation of subscripted matrix variables.

Approximate Time

25 minutes for activity (at home or in class)
45 to 50 minutes for discussion

Classroom Organization

Individuals, followed by whole-class discussion with small group work

Materials

Calculator manuals

Doing the Activity

This activity requires no introduction.

Discussing and Debriefing the Activity

Have students discuss the activity in their groups while you take a quick inventory of how well they did. If most have done a good job, you can go over it quickly.

Based on your impression of students' work, you might go directly to Question 2. Otherwise, ask three students to give their answers to Question 1a to 1c.

Question 2

For Question 2, ask a student to present the matrix of the initial diagram (part a) and another to present the matrix of the translated diagram (part b).

Part c is a key element here. Elicit an explicit equation, which will probably look like this (with perhaps variation in the order of the rows):

$$\begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 3 \\ 3 & 2 \\ 3 & 1 \end{bmatrix} + \begin{bmatrix} 3 & 1 \\ 3 & 1 \\ 3 & 1 \\ 3 & 1 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 2 \\ 4 & 3 \\ 5 & 4 \\ 6 & 3 \\ 6 & 2 \end{bmatrix}$$

Identify the matrix C as the *translation matrix.* Bring out that its rows are all the same and that it simply consists of many copies of the *translation vector* associated with moving three units to the right and one unit up.

Post a rule similar to this one to describe translations:

To translate a matrix of points *a* units in the *x*-direction and *b* units in the *y*-direction, add the matrix $\begin{bmatrix} \boldsymbol{a} & \boldsymbol{b} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \boldsymbol{a} & \boldsymbol{b} \end{bmatrix}$ **to the matrix of points.**

Question 3

Finally, ask students to answer the various parts of Question 3. The point of this question is to acquaint them with standard notation that is similar to the notation their calculators use. Go over the specific notation used by their calculators for matrix entries, which will probably look something like $a(i, j)$ rather than a_{ij}.

Moving the Line with Matrices

Turn to a posted program for *Move That Line!* (one that uses a loop) and ask, **How might we use these ideas about matrices in this program?**

Here is the sample program from the discussion of *Move That Line!*

> Setup program
> Set P equal to 3, Q equal to 2, R equal to 5, and S equal to 1
> For N from 1 to 6
> - Clear the screen
> - Draw the line segment from (P, Q) to (R, S)
> - Delay
> - Add 3 to each of P and R, and add –4 to each of Q and S
>
> End the N loop

If students' programs for *Move That Line!* were quite different from this, you may need to develop a program of this type so that the adaptation to matrices can be done easily.

Bring out that what is involved is merely a change in notation (This isn't meant to minimize the difficulty students may have with the change). Instead of some collection of individual variables for the coordinates (such as P, Q, R, and S), we use one variable: a matrix. Individual entries of that matrix represent the coordinates, with each row of the matrix containing the two coordinates for a different point. Instructions involving the coordinates are replaced by similar matrix instructions. For instance, the initial line giving the values for P, Q, R, and S is replaced by a single instruction defining a matrix.

A complete plain-language program might look like this:

> Setup program
> Let A be the matrix $\begin{bmatrix} 3 & 2 \\ 5 & 1 \end{bmatrix}$
> Let B be the matrix $\begin{bmatrix} 3 & -4 \\ 3 & -4 \end{bmatrix}$
> For N from 1 to 6
> - Clear the screen
> - Draw the line segment from (a_{11}, a_{12}) to (a_{21}, a_{22})
> - Delay
> - Replace matrix A by the sum A + B
>
> End the N loop

This use of matrices may seem like an added complication to students, but if there were many points, it would actually be a simplification. Also, the use of a matrix to represent the two points will provide a convenient mechanism for modifying the points. (In other examples, we will work with more points and so have more rows. Later in the unit, we will have points in 3-space and will need three columns.)

Note: Make a transparency or other written record of this program, and save it to refer to when introducing *Swing That Line!*

Writing the Matrix Program in Code

Ask, **What's the programming code for this plain-language program?** Have students work in groups to create code for this plain-language program (or for their own analogous version from their work on *Move That Line!*). This will require them to review how to define a matrix on their calculators. They will also need to know how to refer to individual entries from a matrix.

They can probably figure this out using their calculator manuals. If some groups seem lost, ask a volunteer from a successful group to tutor them. If no groups figure it out, you may need to pull the class together and go over the procedure.

Have each student enter the program into the calculator to see how it works. Students should write out their matrix programs on paper so they have them to refer to for *Swing That Line!*

Key Questions

How might we use matrices in this program?
What's the programming code for this plain-language program?

Cornering the Cabbage

Intent

In preparation for performing geometric rotations, students develop a formula for the area of a triangle using trigonometry.

Mathematics

This activity will facilitate the discovery that the area of a triangle is half the product of the lengths of two sides times the sine of the included angle. Students will later use that formula to get an expression for the sine of the sum of two angles. They will need the angle-sum formula for calculating rotations of points.

Progression

Students first find the areas of several triangles, given two sides and the included angle. They then generalize their work to find a formula for the area of a triangle and to determine what value for the included angle will maximize the triangle's area. The subsequent discussion extends the generalization to obtuse angles.

Approximate Time

30 minutes for activity (at home or in class)
20 to 30 minutes for discussion

Classroom Organization

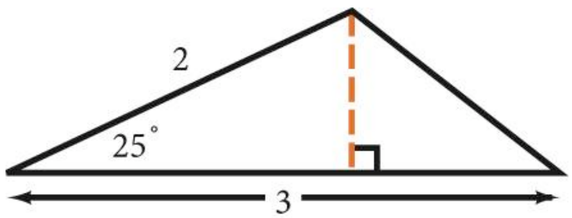
Individuals, followed by whole-class discussion with small group work

Doing the Activity

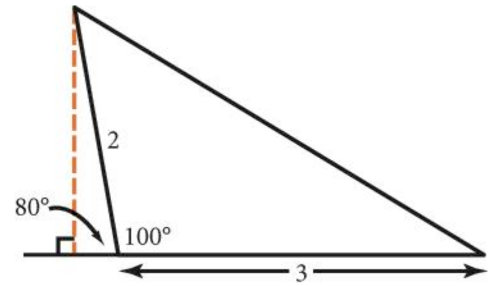
Tell students that although this activity seems unrelated to the material at hand, it will contribute to their ability to do item 3 of the unit outline. You may want to suggest they use something like straws or toothpicks to help visualize the problem.

Discussing and Debriefing the Activity

Ask students to compare what they found for different angles in their groups. Then have someone explain the reasoning used for each part of Question 1. Students will probably use diagrams like these:



Possible diagram for Question 1a



Possible diagram for Question 1b

Though these diagrams use the side of length 3 as the base, students might also set up either diagram using the side of length 2 as the base. For Question 1a, that approach proves a bit more difficult to work with than the diagram here because it leads to an altitude that is outside the triangle. For Question 1b, using either of the sides that form the 100° angle as the base will lead to an altitude that is outside the triangle.

Students should see that the length of the altitude in the first triangle is 2 sin 25° and that the length of the altitude in the second triangle can be written as either 2 sin 100° or 2 sin 80°.

Accept both the 100° and the 80° answers. However, for the purposes of generalizing these examples, it's important that students see that the altitude of the second triangle can be written as 2 sin 100°. Use the occasion to review the general definition of the sine function, having students explain why sin 80° and sin 100° are equal. **Why should sin 80° and sin 100° give the same answer?** Try to elicit explanations both in terms of the formal definition of the sine function and in terms of the Ferris wheel metaphor (from the unit *The Diver Returns*).

Students should then be able to explain how to use the formula $\frac{1}{2}bh$ to get an area of $\frac{1}{2} \cdot 3 \cdot 2 \sin 25°$ for the first triangle and of $\frac{1}{2} \cdot 3 \cdot 2 \sin 100°$ for the second.

(You may want to review the use of the formula $\frac{1}{2}bh$ for the case of an obtuse triangle.) These expressions simplify to 3 sin 25° and 3 sin 100°, respectively, for areas of approximately 1.27 m^2 and 2.95 m^2.

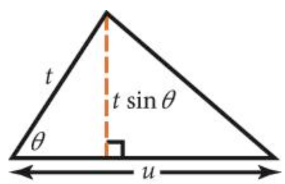
If students are clear on the examples from Question 1, you can probably skip discussion of Question 2.

Question 3

Have a volunteer present Question 3. If no one came up with the general formula $\frac{1}{2}tu \sin \theta$, focus students on the diagram in the activity showing an acute angle for θ. Bring out that the first part of Question 1 is the case of $t = 2$, $u = 3$, and $\theta = 25°$. **What did you do when the angle was 25°?** By having students follow their reasoning in that example, they should be able to come up with the generalization.

Tell students that this rule for calculating the area of the triangle will come in handy soon. Post the general rule, with an appropriate diagram:

> **The area of a triangle is equal to half the product of the length of two sides times the sine of the angle between them.**

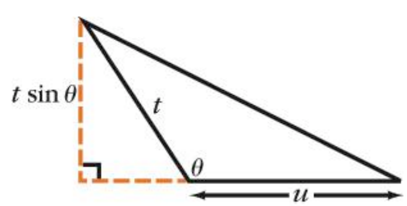


$$\text{Area} = \frac{1}{2}tu \sin \theta$$

Obtuse Angles

If students seemed unclear earlier about the use of the expression 2 sin 100° for the altitude for the second example in Question 1, ask, **Is the area formula $\frac{1}{2}$*tu* sin θ valid for the case where the angle θ between the given sides is more than 90°?**

By using a diagram like the one below and referring to their work with Ferris wheels in *The Diver Returns*, students should be able to justify the fact that the altitude in this diagram is still equal to $t \sin \theta$. For instance, if they view the triangle in a coordinate system with the vertex of angle θ at the origin, then the length of the altitude is the y-coordinate of the other end of the side of length t. Thus, this length is $t \sin \theta$.



Question 4: Maximizing the Area

Have students discuss Question 4. Based on the formula, the diagram, and their intuition, they should see that the best choice is to form a 90° angle with the sticks.

Ask volunteers to present their justifications for this conclusion. They may remain geometrically, from a diagram like the last one, that the altitude will be the biggest when the side of length *t* is vertical. They may also use the fact that the sine function has a maximum value of 1 when the angle is 90°.

Ask, **Why does the sine function have its maximum at 90°?** They might explain this in terms of the formal definition, in terms of the Ferris wheel model, or with a graph.

Key Questions

Why should sin 80° and sin 100° give the same answer?
What did you do when the angle was 25°?

Is the area formula $\frac{1}{2}tu \sin \theta$ valid for the case where the angle θ between the given sides is more than 90°?
Why does the sine function have its maximum at 90°?

Rotation in Two Dimensions

Intent

In these activities, students create programs that include animated rotations.

Mathematics

The technique used here for rotating images will involve matrix multiplication. Developing the rotation matrix requires the use of trigonometric formulas for the sine and cosine of a sum of two angles. Development of those formulas is based in turn upon formulas for the sine and cosine of negative angles, as well as formulas for the sine and cosine of the complement of an angle.

Progression

Goin' Round the Origin introduces the topic of geometric rotations.

Accomplishing rotations with matrices requires the development of a handful of trigonometric identities, which are developed over the course of half a dozen activities. *Double Trouble* hints at the need for a formula for the sine of a sum of two angles. That formula is discovered in *The Sine of a Sum*.

The formula for the cosine of a sum of two angles requires a little more effort to develop. That process begins with identifying a formula for the cosine of the complement of an angle in *A Broken Button*. In *Oh, Say What You Can See*, students review this formula along with two others from an earlier unit that will be needed to find the formula for the cosine of a sum. They will develop further trigonometric identities involving the sine and cosine in *Doubles and Differences*.

The class puts it all together into an identity for the cosine of a sum of two angles and uses the sine-of-a-sum and cosine-of-a-sum formulas to create useful rotation formulas. They put the rotation formulas to work in *Comin' Round Again (and Again...)*.

Students incorporate the rotation formulas into a procedure for using matrices to rotate an image in *More Memories of Matrices*. After pausing to digest this lengthy development process in *How Did We Get Here?*, they try the rotation matrix out in a program in *Swing That Line!* and analyze another program using matrix rotations in *What's Going On Here?*

Goin' Round the Origin
Double Trouble
The Sine of a Sum
A Broken Button

Oh, Say What You Can See
Comin' Round Again (and Again...)
More Memories of Matrices
Taking Steps
How Did We Get Here?
Swing That Line!
Doubles and Differences
What's Going On Here?

Goin' Round the Origin

Intent

This activity introduces rotations as a type of geometric transformation.

Mathematics

Now that students know how to perform the simplest form of isometry, translations, we turn to **rotations.** This activity is a first step toward developing a formula for rotation around the origin. This formula is then expressed in terms of polar coordinates so that it will work in all four quadrants.

Progression

A discussion of where we are in the unit precedes this activity. The class notes that item 3 in the outline developed at the beginning of the unit contains two elements: translations and rotations. The teacher illustrates the idea of rotation with an example. Students then determine the coordinates of two points after specific rotations about the origin and generalize their work to develop a formula.

The subsequent discussion observes that formulas involving the inverse tangent function will not work in all quadrants. This leads to a brief review of polar coordinates and expression of the rotation in terms of polar coordinates.

Approximate Time

15 to 20 minutes for introduction
30 minutes for activity (at home or in class)
40 to 50 minutes for discussion

Classroom Organization

Individuals, preceded and followed by whole-class discussion

Doing the Activity

Have students look at the unit outline from the discussion introducing *Picture This!* They may think they are ready to check off item 3: "Change the position of an object located in a two-dimensional coordinate system." Tell them that they have done part of this—changing the position using a translation—but that their next task involves a different type of position change, a rotation. (The previous activity, *Cornering the Cabbage*, is headed in that direction, although it might seem quite unrelated.)

You may want to modify the outline by indicating two subitems for item 3:

> **3. Change the position of an object located in a two-dimensional coordinate system.**
> - **Translate the object.**
> - **Rotate the object.**

You can then check off the first of these subitems.

Translations in Three Dimensions

Ask students, **How would you represent translations in three dimensions?** Bring out that the basic ideas are the same. To translate a single point, for instance, students would simply need to use a 1-by-3 row vector as the translation vector in place of a 1-by-2 row vector.

Tell students that when they get to three dimensions (item 5 of the unit outline), they will work only with rotations. You might want to emend the outline to reflect this:

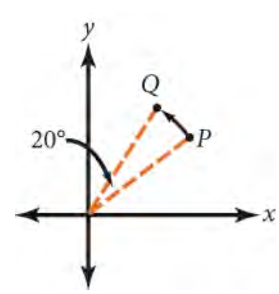
> **5. Change the position of an object located in a three-dimensional coordinate system.**
> - **Rotate the object.**

Rotations: An Introduction

Tell students it is now time to examine how to rotate an object in two dimensions and that they will see over the next few days how the area formula from *Cornering the Cabbage* fits into this task.

Draw a set of coordinate axes and ask students to identify a specific point in the plane (other than the origin). **What would it mean to rotate this point 20° counterclockwise around the origin?** Help students to articulate that rotating the point around the origin means that the distance from the point to the origin will stay fixed and that the counterclockwise angle from the positive x-axis will increase by 20°. Draw in the radii to the origin, so students realize exactly what stays fixed.

For example, a 20° counterclockwise rotation of the point P in the diagram would put it at the position labeled Q.

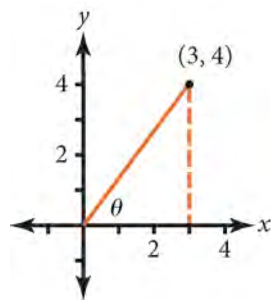


Tell students that in the activity they will be looking for ways to get the coordinates of Q in terms of the coordinates of P and the size of the angle of rotation.

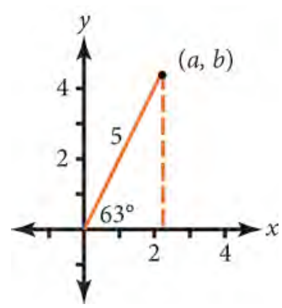
Discussing and Debriefing the Activity

Have a student present Question 1. We describe one likely approach here.

Begin by finding the angle shown as θ in the diagram, perhaps using the condition $\tan\,\theta = \frac{4}{3}$, so $\theta = \tan^{-1}\left(\frac{4}{3}\right)$. This gives $\theta \approx 53.1°$.



Then, using (a, b) to represent the new coordinates as in the next diagram, add 10° to θ, getting approximately 63.1° for the angle shown. Use the fact that the length of the segment from the origin to (a, b) is the same as the length of the segment from the origin to (3, 4). This length is 5, as can be found by the Pythagorean theorem.



Students can then find the coordinates a and b, either from the right triangle or based on the general definition of the sine and cosine functions. Either approach gives $a \approx 5 \cos 63.1° \approx 2.26$ and $b \approx 5 \sin 63.1° \approx 4.46$.

Point out that these are reasonable values, based on the diagram. You may want to save these values for comparison when students develop general formulas for rotation.

Question 2

Use your judgment as to how much detail your class needs in the presentation of Question 2. The coordinates of the new point are approximately (1.10, 5.27). You may also want to save this result for comparison when the general formula is developed.

Question 3

Ask for a volunteer to present the generalization in Question 3. Students should be able to mimic the specific examples to articulate that the new *x*-coordinate, which we'll call x' (read *x prime*), is given by this equation:

$$x' = \sqrt{x^2 + y^2}\cos\left[\tan^{-1}\left(\frac{y}{x}\right) + 10^\circ\right]$$

Similarly, they should find that the new *y*-coordinate, y', is given by this equation:

$$y' = \sqrt{x^2 + y^2}\sin\left[\tan^{-1}\left(\frac{y}{x}\right) + 10^\circ\right]$$

Post these formulas for comparison with another version later in this discussion.

Trouble in the Third Quadrant

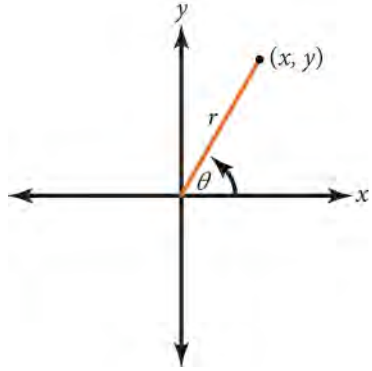
Next, have students try out these formulas with the point (–3, –4). **What do you get when you apply these formulas to try to rotate (–3, –4)?** They should see that the formulas don't work, because they yield the same answer for the point (–3, –4) as for the point (3, 4).

Why don't the formulas work? Let students try to figure out what went wrong. If necessary, remind them that for any number between –1 and 1, there are infinitely many angles whose sine or cosine is that number. A similar situation happens with the tangent function. Therefore, using $\tan^{-1}\left(\frac{y}{x}\right)$ for getting the new angle won't work. Tell students that *Double Trouble* will get them a bit closer to a better set of formulas for x' and y'.

Review of Polar Coordinates

Tell the class there is hope for the development of simpler formulas and that polar coordinates can lead the way. Take this opportunity to review polar coordinates as needed, especially the formulas for finding rectangular coordinates from polar coordinates.

Using a diagram like this one, ask, **How do you express the rectangular coordinates *x* and *y* in terms of the polar coordinates *r* and θ?**



Students should be able to use the right triangle, if necessary, to get the two general formulas:

$$x = r \cos \theta \qquad y = r \sin \theta$$

Post these formulas.

Ask, **Will these formulas work for all values of θ, that is, for all quadrants?** This is an especially important question in the aftermath of the failure of the prior formulas involving the inverse tangent to generalize.

As needed, remind the class that the sine and cosine functions are *defined* in general by the formulas $\cos \theta = \left(\frac{x}{r}\right)$ and $\sin \theta = \left(\frac{y}{r}\right)$, so the formulas for *x* and *y* in terms of *r* and θ hold true for all angles.

Rotation Formulas Using Polar Coordinates

Ask students, **How can you rewrite the formulas for rotations using polar coordinates?** They should recognize the expression $\sqrt{x^2 + y^2}$ as equal to *r* and the expression $\tan^{-1}\left(\frac{y}{x}\right)$ as equal to θ. [In fact, what students need in the formula is *always* θ, and θ is only *sometimes* equal to $\tan^{-1}\left(\frac{y}{x}\right)$.] Thus, they should see that if the original point has polar coordinates (r, θ), then the result of rotating 10° is the point (x', y'), where

$$x' = r \cos(\theta + 10°) \qquad y' = r \sin(\theta + 10°)$$

and where $r = \sqrt{x^2 + y^2}$.

Post this new pair of formulas for x' and y' for comparison with the earlier versions.

Will these new formulas work for all values of θ? Students should see that they do, but that to use these new formulas productively, they seem to need two things:

- A good expression for θ in terms of x and y
- A way to handle the sine and the cosine of a sum

Tell students that with the help of the next activity, *Double Trouble*, they will develop a way to deal with the second of these items and at the same time eliminate the need for the first. That will allow them to avoid the use of the inverse tangent function, which doesn't necessarily give the desired value outside the first quadrant.

Key Questions

How would you represent translations in three dimensions?
What would it mean to rotate this point 20° around the origin?
What do you get when you apply these formulas to try to rotate (–3, –4)?
Why don't the formulas work?
How do you express the rectangular coordinates x and y in terms of the polar coordinates r and θ?
Will these formulas work for all values of θ, that is, for all quadrants?
How can you rewrite the formulas for rotations using polar coordinates?
Will these new formulas work for all values of θ?

Double Trouble

Intent

This activity will ease students into discovering the formula for the sine of a sum, which will be the key for expressing rotations.

Mathematics

Students develop a formula for the sine of twice an angle.

Progression

Students first consider whether sin 80° is equal to 2 sin 40°. They then develop an expression equivalent to sin 80° by bisecting the angle at the apex of a triangle and using the formula for the area of a triangle that was developed in *Cornering the Cabbage* to set the area of the original triangle equal to the sum of the areas of the two smaller triangles. This yields the relationship sin 80° = 2 sin 40° cos 40°.

Approximate Time

25 minutes for activity (at home or in class)
15 minutes for discussion

Classroom Organization

Individuals, followed by whole-class discussion

Doing the Activity

No introduction of the activity is necessary.

Discussing and Debriefing the Activity

Ask a student to summarize Woody's problem. His dilemma is that he knows the values of sin 40° and cos 40°, but wants to find the value of sin 80°.

Then turn to Question 1. **Can Woody simply multiply sin 40° by 2 to get sin 80°?** Presumably, students found that sin 80° and 2 sin 40° are not equal, probably by evaluating both expressions on a calculator.

Then move on to Question 2. **How can you show that sin 80° and 2 sin 40° are unequal without using a calculator?**

If no one has any suggestions, ask, **What is 2 sin 40°? Why can't sin 80° be equal to this value?** Students can find 2 sin 40° using the fact that sin 40° is .643. Because 2 sin 40° is more than 1, it can't be the sine of any angle.

Are there angles whose sine you know without using a calculator? If necessary, suggest students think about the case of 90° and 180°. They should see that sin 180° is not equal to 2 sin 90°.

Bring out that the condition "sin 180° ≠ 2 sin 90°" does not prove that sin 80° is not equal to 2 sin 40°, but it does show that the equation sin $2A = 2 \sin A$ does not hold true for all angles. And as students have just seen, it cannot hold true if sin A is greater than .5.

Ask, **How can you use the Ferris wheel model to see why sin 80° ≠ 2 sin 40°?** Try to elicit the explanation that going 80° around the Ferris wheel (from the 3 o'clock position) does not place the rider twice as far above the center as going 40° around the Ferris wheel.

You may want to ask students to check other examples on their calculators. You can also have them see from the graph of the sine function that doubling the x-value does not lead to doubling the y-value.

Question 3

Next, ask other students for expressions for the areas of the large triangle and the smaller triangles. They will probably give these equations:

$$\text{Area of large triangle} = \frac{1}{2}x^2 \sin 80°$$

$$\text{Area of each smaller triangle} = \frac{1}{2}xz \sin 40°$$

Question 4

Ask for a volunteer to answer Question 4a. If no one got an answer, have students work on this in their groups, perhaps asking how the areas of the triangles are related. From the fact that the area of the large triangle is the sum of the areas of the two smaller triangles, they should get the equation

$$\frac{1}{2}x^2 \sin 80° = 2 \cdot \frac{1}{2}xz \sin 40°$$

Multiplying both sides by 2, dividing both sides by x^2, and simplifying leads to

$$\sin 80° = 2 \cdot \frac{z}{x} \cdot \sin 40°$$

Next, turn to Question 4b. Again, if necessary, let students work in their groups to "get rid of x and z" or to "bring cos 40° into the equation." You might remind them that Woody knows cos 40°, or else ask, **What does the ratio $\frac{z}{x}$ represent within the small right triangle?** This should lead them to the formula

$$\sin 80° = 2 \cos 40° \sin 40°$$

Connect this formula with work so far on rotations by writing this as

$$\sin (40° + 40°) = 2 \cos 40° \sin 40°$$

Point out, if necessary, that this equation involves finding the sine of a sum of two angles, as do the rotation formulas using polar coordinates, which involve sin (θ + 10°).

Let students speculate briefly about whether the last equation is something peculiar to 40° or whether it generalizes. Then tell them that in the next activity, they will generalize this idea to a case in which the two angles being added are not equal.

Key Questions

Can Woody simply multiply sin 40° by 2 to get sin 80°?
How can you show that sin 80° and 2 sin 40° are unequal without using a calculator?
What is 2 sin 40°? Why can't sin 80° be equal to this value?
Are there angles whose sine you know without using a calculator?
How can you use the Ferris wheel model to see why sin 80° ≠ 2 sin 40°?

What does the ratio $\frac{z}{x}$ represent within the small right triangle?

The Sine of a Sum

Intent

Students discover a formula for the sine of the sum of two angles.

Mathematics

The formulas for the coordinates of a point after rotation about the origin that the class developed in the discussion of *Goin' Round the Origin* involve finding the sine and cosine of a sum of two angles. This activity focuses on the sine of the sum. Students discover that $\sin(A + B) = \sin A \cos B + \cos A \sin B$.

Progression

Students use a variation of the triangle from *Cornering the Cabbage* to develop a formula for $\sin(A + B)$. In the follow-up discussion, they verify the formula using specific cases.

Approximate Time

55 to 65 minutes

Classroom Organization

Small groups or individuals, followed by whole-class discussion

Doing the Activity

Groups can begin the activity without an introduction.

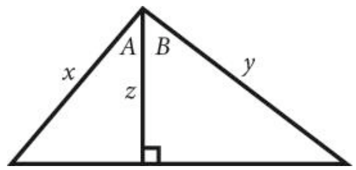
Discussing and Debriefing the Activity

As groups conclude their work, ask one or two of them to prepare presentations.

Before beginning the presentations, ask, **Why might you want a formula for the sine of a sum?** If needed, point to the posted formulas for rotations that involve the expressions $\cos(\theta + 10°)$ and $\sin(\theta + 10°)$.

Developing the Formula

If students use the diagram in the activity (and shown below), they should see that the area of the large triangle is $\frac{1}{2}xy \sin(A + B)$ and that the two smaller triangles have areas $\frac{1}{2}xz \sin A$ and $\frac{1}{2}yz \sin B$.



This should lead to the equation

$$\frac{1}{2}xy \sin(A + B) = \frac{1}{2}xz \sin A + \frac{1}{2}yz \sin B$$

One approach from here is to multiply both sides by 2, divide by *xy,* and simplify. (If needed, refer students to their similar work on *Double Trouble.*) This gives

$$\sin(A + B) = \frac{z}{y}\sin A + \frac{z}{x}\sin B$$

But the ratios $\frac{z}{y}$ and $\frac{z}{x}$ are simply $\cos B$ and $\cos A$ respectively, so the equation becomes

$$\sin(A + B) = \cos B \sin A + \cos A \sin B$$

Post this formula prominently and label it "Sine of a Sum."

Point out that the argument used here proves the relationship only for acute angles, but that the formula holds true for all angles. Let students spend a few minutes verifying with their calculators that the formula holds true for angles in all quadrants. (The supplemental activities *Moving to the Second Quadrant* and *Sums for All Quadrants* involve proving this equation more generally.)

Verifying Special Cases

Have students check some special cases, such as $B = 0°$, $90°$, and $180°$. Help them see (using the specific values of sine and cosine for these angles) that in these cases, the sine-of-a-sum formula simplifies to these equations:

- $\sin A = \sin A$

- sin (A + 90°) = cos A
- sin (A + 180°) = −sin A

The first is, of course, true for all values of *A.* You might suggest that students use the graphs of the sine and cosine functions to develop an explanation for the second result.

The third identity is considered in the supplemental activity *Adding 180°*. Students may be able to give a good explanation at this point in terms of the graph or the Ferris wheel model.

Also ask, **What does the sine-of-a-sum formula say if *B* is equal to *A*?** Students should see that it then becomes the "double-angle" formula

$$\sin 2A = 2 \cos A \sin A$$

and that this is consistent with students' results in *Double Trouble,* namely

$$\sin 80° = 2 \cos 40° \sin 40°$$

Key Questions

Why might you want a formula for the sine of a sum?
What does the sine-of-a-sum formula say if *B* is equal to *A*?

A Broken Button

Intent

Students develop relationships between sine and cosine.

Mathematics

This activity will pave the way for finding a formula for the cosine of a sum in the discussion introducing *Comin' Round Again* (*and Again...*). Students use the Pythagorean identity, $\sin^2 x + \cos^2 x = 1$, to discover the identity $\cos\theta = \sin(90^\circ - \theta)$.

Progression

Students use the Pythagorean identity and the numeric value of $\sin 50^\circ$ to find the value of $\cos 50^\circ$. They then look at how they might find $\cos 50^\circ$ by finding the sine of a different angle in a right triangle. The subsequent discussion confirms that the cosine of an angle is equal to the sine of its complement.

Approximate Time

25 minutes for activity (at home or in class)
20 minutes for discussion

Classroom Organization

Individuals, then groups, followed by whole-class discussion

Doing the Activity

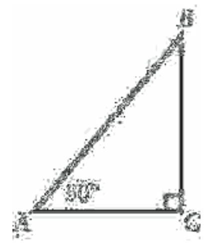
This activity requires no introduction.

Discussing and Debriefing the Activity

Give students a few minutes to share ideas on Questions 1 and 2 in their groups. Then have students from two groups present results on the two questions.

For Question 1, using the Pythagorean identity, one would get $\sin^2 50° + \cos^2 50° = 1$. Because Woody knows the value of $\sin 50°$ (approximately .766), he can substitute to get $.766^2 + \cos^2 50° = 1$, so $\cos^2 50° \approx .413$ and $\cos 50° \approx .643$.

For Question 2, students may have seen immediately that cos 50° is the same as sin 40°. If not, or if you feel a full discussion is needed, have students explain their reasoning using a right triangle like this one.



They should see that on the one hand, cos 50° is equal, by definition, to the ratio $\frac{AC}{AB}$. On the other hand, this ratio is also the sine of $\angle CBA$, so $\angle CBA = 40°$ and cos 50° = sin 40°.

Generalizing the Result

Ask students how 40° and 50° are related to each other. **What is the relationship between 40° and 50°? What are such angles called?** Review that a pair of angles whose sum is 90° are called *complementary angles*.

Can you generalize the equation you found for 50°, cos 50° = sin 40°? Students should come up with the formula

$$\cos\theta = \sin(90° - \theta)$$

and see that this makes sense for any right triangle with an acute angle of θ.

Ask students whether the identity also works where the right-triangle reasoning falls apart. **Does $\cos\theta = \sin(90° - \theta)$ also hold true for nonacute angles?** They may choose to test this with values of θ from various quadrants, or they may try to reason based on the graphs of the functions. (The latter idea is pursued in *Oh, Say What You Can See*. Also, it is important for the unit problem that students accept this general formula. It is not essential that they know the details of a proof.)

If students did not already use the formula $\sin\theta = \cos(90° - \theta)$, ask if there is a formula similar to $\cos\theta = \sin(90° - \theta)$ that Woody could have used if the buttons had been reversed—that is, if his sine key had been broken and he wanted to find, say, sin 50° using his cosine key. This should lead them to develop the variation.

Post both of these formulas. Students will use them in the development of the formula for the cosine of the sum of two angles, which introduces *Comin' Round Again (and Again...)*.

$$\cos \theta = \sin (90° - \theta) \qquad \sin \theta = \cos (90° - \theta)$$

This is a good opportunity to suggest that students keep a list of trigonometric identities. They will be using known identities to develop new ones, so it will be helpful to have a single place where they can find this material.

Key Questions

What is the relationship between 40° and 50°? What are such angles called?

Can you generalize the equation you found for 50°, cos 50° = sin 40°?

Does cos θ = sin (90° − θ) also hold true for nonacute angles?

Oh, Say What You Can See

Intent

Students justify various formulas involving sine and cosine.

Mathematics

Students developed the identity $\cos \theta = \sin (90^\circ - \theta)$ in *A Broken Button*. Now they justify that formula and two others, $\cos (-\theta) = \cos \theta$ and $\sin (-\theta) = -\sin \theta$, using the Ferris wheel model or the graphs of the sine and cosine functions.

Progression

Students test the equation $\cos \theta = \sin (90^\circ - \theta)$ in multiple quadrants and then explain it both in terms of the Ferris wheel model and in terms of the graphs of the sine and cosine functions. They do the same with the formulas for the sine and cosine of a negative angle, explaining them using either the Ferris wheel or the graphs.

Approximate Time

30 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

Individuals, followed by whole-class discussion

Materials

Optional: Transparency of *Oh, Say What You Can See* blackline master

Doing the Activity

Students work on this activity independently.

Discussing and Debriefing the Activity

For Question 1, have students share explanations of the equation $\cos \theta = \sin (90° - \theta)$ in terms of the Ferris wheel and the graph. Focus especially on why this equation holds true for all angles (and not merely for acute angles).

This is also a good opportunity to look at this equation in terms of the formal definition of sine and cosine. You might point out that when a point in the coordinate plane is rotated 90° about the origin, its x- and y-coordinates are essentially interchanged, except for the signs. (One might view the angle $90° - \theta$ as the result of a reflection of the angle θ through the x-axis followed by a 90° counterclockwise rotation.)

It's important that students feel fairly comfortable with the formulas $\cos(-\theta) = \cos\theta$ and $\sin(-\theta) = -\sin\theta$, as they will use them in the discussion introducing the next activity to transform the formula for the sine of a sum into a formula for the cosine of a sum. Focus students' attention on how these relationships are illustrated in the graphs of the two functions.

You might also want to post the trigonometric identities in this activity and use this occasion to review ideas about the general definitions of sine and cosine. As noted in the discussion of *A Broken Button*, you may want to have students keep individual lists of identities as well.

Comin' Round Again (and Again...)

Intent

Students develop and apply a formula for the cosine of a sum of two angles.

Mathematics

The class now discovers and justifies a formula for the cosine of the sum of two angles. This is used together with the sine-of-a-sum formula to simplify the rotation formulas.

Progression

The teacher introduces this activity by leading students to derive a formula for the cosine of a sum from previously derived formulas. The class verifies the formula in special cases. The teacher then leads students in using this new formula to derive formulas for the coordinates of rotated points in terms of the original coordinates and the angle of rotation.

The activity is basically an exercise in applying the new formulas, where students draw a picture on graph paper, use the formulas to rotate each endpoint, and then plot the rotated points to draw the rotated picture. No subsequent discussion is necessary. Students should save their results for later use.

Approximate Time

40 to 45 minutes for introduction
30 to 40 minutes for activity (at home or in class)

Classroom Organization

Whole-class discussion interspersed with small group work, followed by individuals

Doing the Activity

Tell students they are now ready to find the cosine of a sum of two angles, building on their work from *The Sine of a Sum*. Although you might want to let groups try to do this on their own, this formula might best be developed primarily through whole-class discussion. You can ask questions and offer hints to guide the class along, occasionally having groups work on pieces of the task.

To start, be sure all of these formulas are readily available:

- $\sin(A + B) = \sin A \cos B + \cos A \sin B$
- $\cos \theta = \sin(90° - \theta)$
- $\sin \theta = \cos(90° - \theta)$
- $\cos(-\theta) = \cos \theta$
- $\sin(-\theta) = -\sin \theta$

Tell students, **You will be putting these five formulas together to create an expression for cos ($A + B$).** You might preface this work by asking if anyone has questions about any of these formulas.

The first step is for students to relate the task of developing a cosine-of-a-sum formula to what they already know about the sine of a sum. Toward this end, ask, **What is the expression for which we are trying to find a formula? What do we want on the left side of the equal sign?** Students should see that they are trying to find an expression for $\cos(A + B)$ in terms of the individual sines and cosines of A and B.

Which of the five formulas could you use to get an equivalent expression for cos ($A + B$)? If needed, ask how they might rewrite this expression in terms of sine (for which they have a "sum" formula). This should lead to the idea of using the equation $\cos \theta = \sin(90° - \theta)$, with $A + B$ in place of θ, and thus to the equation

$$\cos(A + B) = \sin[90° - (A + B)]$$

Tell students that to use the sine-of-a-sum formula, they need to write the right side of this equation as the sine of a sum. **How could you rewrite the expression 90° − ($A + B$) as a sum?** First suggest they write $90° - (A + B)$ simply as $90° - A - B$, so the chain of development should now look like this:

$$\begin{aligned}\cos(A + B) &= \sin[90° - (A + B)] \\ &= \sin(90° - A - B)\end{aligned}$$

If students don't come up with the idea of writing $90° - A - B$ as $(90° - A) + (-B)$, simply point out that these two expressions are equal. The chain of development should now look like this:

$$\begin{aligned}\cos(A + B) &= \sin[90° - (A + B)] \\ &= \sin(90° - A - B) \\ &= \sin[(90° - A) + (-B)]\end{aligned}$$

What happens if you apply the sine-of-a-sum formula to the right side? This should lead students to develop the last line in this sequence:

$$\begin{aligned}\cos(A+B) &= \sin[90° - (A+B)]\\ &= \sin(90° - A - B)\\ &= \sin[(90° - A) + (-B)]\\ &= \sin(90° - A)\cos(-B) + \cos(90° - A)\sin(-B)\end{aligned}$$

Next, ask students to simplify this expression by using the last four items in the list of five formulas at the beginning of this discussion. This should produce these equations:

- $\sin(90° - A) = \cos A$
- $\cos(-B) = \cos B$
- $\cos(90° - A) = \sin A$
- $\sin(-B) = -\sin B$

Substituting the right sides of these equations into the last line of the chain of development so far will produce this sequence:

$$\begin{aligned}\cos(A+B) &= \sin[90° - (A+B)]\\ &= \sin(90° - A - B)\\ &= \sin[(90° - A) + (-B)]\\ &= \sin(90° - A)\cos(-B) + \cos(90° - A)\sin(-B)\\ &= \cos A\cos B - \sin A\sin B\end{aligned}$$

Put the beginning and end of this chain together as "Cosine of a Sum," and post this formula along with the sine-of-a-sum formula:

$$\mathbf{\cos(A+B) = \cos A\cos B - \sin A\sin B}$$

Why Is This Formula Correct?

Ask students, **How do you know this formula holds true for all angles?** If need be, point out that they derived this formula from formulas they had previously proved (at least for acute angles). Thus, their derivation of this formula is a proof of it, subject to some restrictions.

Clarify that they proved the sine-of-a-sum formula only for acute angles (unless they have done the supplemental activities *Moving to the Second Quadrant, Adding 180°,* and *Sums for All Quadrants*). So they have only proved this cosine formula for some angles. But assure them that it is true for all angles, and that if time and patience permitted, they could prove it for all angles. The other formulas involved in the proof have been proved for all angles.

Verifying Special Cases

As with the sine-of-a-sum formula, have students check some special cases, such as $B = 0°$, $90°$, and $180°$. Help students see (using the specific values of sine and cosine for these angles) that in these cases, the cosine-of-a-sum formula simplifies to these equations:

- $\cos A = \cos A$
- $\cos (A + 90°) = -\sin A$
- $\cos (A + 180°) = -\cos A$

The first is, of course, true for all values of *A.* You might suggest that students use the graphs of the sine and cosine function to develop an explanation for the second result.

The third of these identities is considered in the supplemental activity *Adding 180°*. Students may be able to give a good explanation at this point in terms of the graph or the Ferris wheel model.

Rotations Revisited

Ask students, **Why are we interested in these formulas?** As a hint, ask, **Where do you stand so far in developing formulas for the result of the rotation of a point?** They should be able to point to these two formulas in terms of r and θ:

$$x' = r \cos (\theta + 10°)$$
$$y' = r \sin (\theta + 10°)$$

Have students work in groups to apply their newfound formulas for sine-of-a-sum and cosine-of-a-sum to the formulas for x' and y'. They should get something like this:

$$x' = r (\cos \theta \cos 10° - \sin \theta \sin 10°)$$
$$y' = r (\sin \theta \cos 10° + \cos \theta \sin 10°)$$

Next, ask, **What happens if you distribute the *r* through the expression?** This should produce these equations:

$$x' = r \cos \theta \cos 10° - r \sin \theta \sin 10°$$
$$y' = r \sin \theta \cos 10° + r \cos \theta \sin 10°$$

Where have you seen *r* cos *θ* before? Using the relationships $y = r \sin \theta$ and $x = r \cos \theta$, students should be able to rewrite the formulas as

$$x' = x \cos 10° - y \sin 10°$$
$$y' = y \cos 10° + x \sin 10°$$

To make the second equation easier to work with in terms of the matrix representation, you may want to rewrite it to put the term with *x* first:

$$y' = x \sin 10° + y \cos 10°$$

With a bit of fanfare, point out that students have eliminated *r* and θ from the formula and written the new *x*- and *y*-coordinates directly in terms of the original *x*- and *y*-coordinates.

You may want to have students check this formula for the points (3, 4) and (2, 5) (assuming you saved the answers to *Goin' Round the Origin*) to see that it gives the right answers.

Before moving on, ask, **Is there anything special about 10°?** Presumably, students will be comfortable with replacing this by a general value. To avoid confusion, you might want to introduce a Greek letter other than θ for the angle of rotation. We will use the letter ϕ, which is called *phi* (pronounced to rhyme with *pea*).

Post this general statement:

> **If a point (x, y) is rotated counterclockwise through an angle ϕ, then the coordinates (x', y') of its new location are given by the equations**
>
> $$x' = x \cos \phi - y \sin \phi$$
> $$y' = x \sin \phi + y \cos \phi$$

Verifying Special Cases

Once again, it's worthwhile to have students investigate whether these formulas make sense for some special cases. In particular, ask, **What do these formulas say if $\phi = 0°$? If $\phi = 90°$? If $\phi = 180°$?**

Discussing and Debriefing the Activity

After the introduction above, this should be a fairly routine assignment and a good indicator of whether students are comfortable with the work in developing the rotation formulas. You need not take time to go over the activity unless there are questions. However, ask students to save their results. If time allows, they will compare them with an approach to rotations using matrices following *More Memories of Matrices*.

Key Questions

What do we want on the left side of the equal sign?
Which of the five formulas could you use to get an equivalent expression for cos ($A + B$)?
How could you rewrite the expression 90° − ($A + B$) as a sum?
What happens if you apply the sine-of-the-sum formula to the right side?
How do you know this formula holds true for all angles?
What happens if you distribute the r through the expression?
Where have you seen $r \cos \theta$ before?
What do these formulas say if $\phi = 0°$? If $\phi = 90°$? If $\phi = 180°$?

Supplemental Activities

Sum Tangents **(reinforcement)** asks students to prove the formula for the tangent of a sum of two angles.
Moving to the Second Quadrant **(reinforcement)** asks students to prove the formula for the sine of the sum of two angles for the case in which one of the angles is in the first quadrant and the other is in the second quadrant.
Adding 180° **(reinforcement)** asks students to develop formulas for $\sin(\theta + 180°)$ and $\cos(\theta + 180°)$.
Sums for All Quadrants **(extension)** challenges students to prove the sine-of-a-sum formula for all angles.
Polar Complex **(extension)** has students use the formulas for the sine and cosine of a sum of two angles to explore the product and quotient of two complex numbers written in polar form.
Sine and Cosine Derivatives Again **(extension)** asks students to use the formulas for the sine and cosine of a sum of two angles in another approach to developing formulas for the derivative of the sine and cosine functions.

More Memories of Matrices

Intent

Students use matrix multiplication to express rotations.

Mathematics

This activity reviews matrix multiplication and then applies it to rotations. This is the last piece in developing a nice formulation for rotations.

Progression

The activity opens with a problem situation that will remind students how to perform matrix multiplication. After students generalize by describing that process, they're reminded that matrix multiplication is not commutative. They practice matrix multiplication with one more problem and then use the idea of matrix multiplication to express rotations. In the subsequent discussion, students apply rotation matrices to their work from *Comin' Round Again* (*and Again...*) after confirming the general formula for the rotation matrix.

Approximate Time

85 minutes

Classroom Organization

Small groups or individuals, followed by whole-class discussion

Doing the Activity

In Questions 1 to 4, students will review multiplication of matrices and then use matrix multiplication for rotation of points. If they get stuck at Question 5, you may want to tell them explicitly to look for a 2×2 matrix M such that

$$\begin{bmatrix} x & y \end{bmatrix} \cdot \begin{bmatrix} \mathrm{M} \end{bmatrix} = \begin{bmatrix} x\cos\phi - y\sin\phi & x\sin\phi + y\cos\phi \end{bmatrix}$$

Discussing and Debriefing the Activity

Have a student present all of Question 1. The directions are fairly explicit about how to arrange the information, so students should come up with these matrices:

$$[A] = \begin{bmatrix} 40 & 2 \\ 50 & 3 \end{bmatrix}$$
$$[B] = \begin{bmatrix} 500 & 200 \end{bmatrix}$$
$$[C] = \begin{bmatrix} 30{,}000 & 1{,}600 \end{bmatrix}$$

The matrix multiplication involved in Question 1c is this equation:

$$\begin{bmatrix} 500 & 200 \end{bmatrix} \cdot \begin{bmatrix} 40 & 2 \\ 50 & 3 \end{bmatrix} = \begin{bmatrix} 30{,}000 & 1{,}600 \end{bmatrix}$$

Ask students to explain which numbers are multiplied and what gets added to get 30,000 and 1,600. **How exactly did you get 30,000 and 1,600?** It may be helpful to explicitly write out equations like these:

$$500 \cdot 40 + 200 \cdot 50 = 30{,}000$$
$$500 \cdot 2 + 200 \cdot 3 = 1{,}600$$

Question 2

Ask two or three students for their descriptions of how to multiply matrices.

Question 3

The matrix multiplication here is

$$\begin{bmatrix} 40 & 2 \\ 50 & 3 \end{bmatrix} \cdot \begin{bmatrix} 500 & 200 \end{bmatrix}$$

which does not make sense. Review the fact that not every pair of matrices can be multiplied, and go over the conditions on the dimensions that make matrix multiplication possible. The number of columns in the first matrix must be equal to the number of rows in the second matrix.

What does this show about matrix multiplication? Emphasize that this shows that matrix multiplication is not commutative.

Question 4

Have another student present Question 4. Matrix D should look like this:

$$[D] = \begin{bmatrix} 500 & 200 \\ 400 & 300 \\ 250 & 350 \end{bmatrix}$$

The presenting student should explain that the matrix of weights and volumes can be found as the product [D] · [A]:

$$\begin{bmatrix} 500 & 200 \\ 400 & 300 \\ 250 & 350 \end{bmatrix} \cdot \begin{bmatrix} 40 & 2 \\ 50 & 3 \end{bmatrix} = \begin{bmatrix} 30{,}000 & 1{,}600 \\ 31{,}000 & 1{,}700 \\ 27{,}500 & 1{,}550 \end{bmatrix}$$

Use your judgment about whether to get an explicit explanation of which numbers were multiplied together, but be sure students can explain the meaning of the entries in the product matrix.

Question 5

If many groups were stuck by the question of how to use matrices to represent the process of rotation of points, you may want to let them return to it now, with perhaps a clearer idea about matrix multiplication than they had previously.

If needed, discuss the form of the equation. Students are looking for a matrix M for which the product $\begin{bmatrix} x & y \end{bmatrix} \cdot \begin{bmatrix} \mathrm{M} \end{bmatrix}$ comes out to $\begin{bmatrix} x\cos\phi - y\sin\phi & x\sin\phi + y\cos\phi \end{bmatrix}$. They should see that they want

$$\begin{bmatrix} \mathrm{M} \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}$$

In other words, they can find the new coordinates using this matrix multiplication:

$$\begin{bmatrix} x & y \end{bmatrix} \cdot \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} = \begin{bmatrix} x\cos\phi - y\sin\phi & x\sin\phi + y\cos\phi \end{bmatrix}$$

Introduce the term *rotation matrix* for the matrix $\begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}$.

Rotating Many Points

Finally, as a follow-up, ask, **How could you use matrix multiplication to rotate several points together?** You might remind students that in *Memories of Matrices,* they put the coordinates for many points in a two-column matrix, with a separate row for each point.

If they need assistance, have them set up a matrix of initial points, either numerically or using variables, to make the question clear. For example, you might suggest they start with three points, (x, y), (s, t), and (u, v), and make a 3×2 matrix from them:

$$\begin{bmatrix} x & y \\ s & t \\ u & v \end{bmatrix}$$

What would the final positions of each of these points be if they were all rotated through an angle ϕ? Students should see that the desired matrix is

$$\begin{bmatrix} x\cos\phi - y\sin\phi & x\sin\phi + y\cos\phi \\ s\cos\phi - t\sin\phi & s\sin\phi + t\cos\phi \\ u\cos\phi - v\sin\phi & u\sin\phi + v\cos\phi \end{bmatrix}$$

Finally, they need to see that they can get this matrix for the rotated points by multiplying the 3×2 matrix on the right by [M], just as they did for a single point.

This result deserves a lot of fanfare and should be posted:

> **To rotate a matrix of two-dimensional points counterclockwise around the origin by an angle ϕ, multiply that matrix on the right by the matrix**
>
> $$\begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}$$

Students should also have this idea in their notes for use on later activities.

Applying Matrix Rotations to Specific Examples

To reinforce the method and strengthen students' faith in it, have them test the process with one or more specific examples. Let the class choose the endpoints for a segment in the first quadrant and draw the segment on a coordinate grid.

For simplicity, ask students to predict where each point would end up if rotated counterclockwise around the origin through an angle of 90°. They should multiply by the appropriate matrix and find out.

If time allows, have students work in groups using the matrix multiplication process to check their answers on *Comin' Round Again* (*and Again...*).

Key Questions

How exactly did you get 30,000 and 1,600?
Do you have any mental images that help you remember how to multiply matrices?
What does this show about matrix multiplication?
How could you use matrix multiplication to rotate several points together?

Supplemental Activity

***Bugs in Trees* (extension)** gives students more practice using matrices.

Taking Steps

Intent

Students analyze variations in the use of the For/End instructions.

Mathematics

This activity, which explores the effects of incrementing the loop variable by something other than 1, is a programming sidelight that students may find useful in their final projects.

Progression

Students analyze and write small programs that use a loop step value of other than 1.

Approximate Time

25 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

Individuals, followed by whole-class discussion

Doing the Activity

Students work on this activity independently.

Discussing and Debriefing the Activity

Ask two students to provide answers to Questions 1 and 2. For Question 1, they will probably agree that the calculator will print the numbers 2, 5, 8, and 11, and then stop. For Question 2, they are likely to mimic the program in Question 1, using instructions like these:

```
For T from 30 to 50, step 5
•    Show T on the screen
End the T loop
```

For Question 3, students will have had to guess a bit, because there is more than one reasonable way a calculator might handle these variations.

In effect, most calculators increase the loop variable each time the program comes back to the beginning of the loop (to the “For” instruction). If this increase makes the loop variable greater than the final value (10, in the case of Question 3a), the calculator considers the loop finished and goes directly to the instruction that comes after the end of the loop, that is, after the “End” instruction. (In this structure, after a loop has been completed, the loop variable is inevitably larger than the final value, or smaller if the increment is negative.)

In other words, the result of the program in Question 3a is that the calculator displays the values 2, 5, and 8 only. For Question 3b, it displays 3, 3.7, 4.4, 5.1, and 5.8. Have students check that this is how their calculators operate.

Students’ work on Question 4 will also have been speculative. If nobody came up with the idea of using a negative increment, leave this as an open question for students to investigate in their spare time. Many calculators do allow negative increments, with the initial value larger than the final value.

How Did We Get Here?

Intent

Students reflect on what they have learned so far in the unit.

Mathematics

Students will review all the key mathematical concepts covered in the unit thus far.

Progression

In this activity, students state the purpose of the unit, list the mathematical terms and formulas they have learned, and discuss how each one relates to the unit's purpose. The subsequent discussion includes a review of the development of the rotation formula.

Approximate Time

5 minutes for introduction
30 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

Individuals, then groups, preceded and followed by whole-class discussion

Doing the Activity

Take a few minutes to have the class brainstorm a list of mathematical terms that students have worked with in this unit.

Discussing and Debriefing the Activity

Have students confer in their groups. Then ask a member of each group to suggest things they have learned.

Next, have one or two students describe what they see as the purpose of the unit. Students from other groups can critique the statements until a formulation is reached that most of the class is happy with. Then have several students discuss how each item in the list of things learned relates to the purpose of the unit.

Also use this discussion to focus on the development of the rotation formula. Ask students about this series of activities—specifically, how they relate to one another and how each contributed to the development of the formula:

- *Cornering the Cabbage*
- *Goin' Round the Origin*
- *Double Trouble*
- *The Sine of a Sum*
- *A Broken Button*
- *Oh, Say What You Can See*
- The "cosine of a sum" discussion introducing *Comin' Round Again* (*and Again...*)

Swing That Line!

Intent

Students use rotation matrices in a calculator program.

Mathematics

Students apply their new rotation formulas to calculator graphics as they write an animation program to rotate a line segment around the origin.

Progression

The teacher introduces the activity by reviewing programs from *Move That Line!* and the generic animation outline. Students then sketch a line segment rotating around the origin in increments of a fixed number of degrees. They write a plain-language program for accomplishing this animation, turn it into programming code, test it in their calculators, and then embellish it to work with a more complicated picture. Pairs share their results with the class.

Approximate Time

70 to 75 minutes

Classroom Organization

Pairs, preceded by whole-class discussion and followed by several presentations

Materials

A copy of the program to move a line using matrices (see the discussion notes for *Memories of Matrices*)

Doing the Activity

Show students the program they made using matrices for *Move That Line!* Use the transparency or chart paper copy of this program from the discussion following *Memories of Matrices.* Students should have also saved a copy of their own programs. Here is the program as described in the *Memories of Matrices* discussion notes:

Setup program

Let A be the matrix $\begin{bmatrix} 3 & 2 \\ 5 & 1 \end{bmatrix}$

Let B be the matrix $\begin{bmatrix} 3 & -4 \\ 3 & -4 \end{bmatrix}$

For N from 1 to 6
- Clear the screen
- Draw a line from (a_{11}, a_{12}) to (a_{21}, a_{22})
- Delay
- Replace matrix A by the sum A + B

End the N loop

Ask for volunteers to describe what is happening in this program. One thing they should note is that the endpoints of the segment are put into a matrix called [A] and that these endpoints are later translated by the process of adding a special *translation matrix* to [A] (in the "Replace" line of the loop). This sum then becomes the new value of [A] and contains the endpoints of the translated segment.

You may also want to have students review the "generic" animation program outline they wrote as a class while discussing the results from *Move That Line!* The outline given there looked like this:

Setup program
Set the initial coordinates
Start the loop
- Clear the screen
- Draw the next figure
- Delay
- Change the coordinates

End the loop

Tell students their next task is to write a program with a similar structure to *rotate* a line segment. As before, they will store the endpoints in a matrix.

Ask students, **What else did you set at the beginning of the *Move That Line!* program, besides the initial coordinates?** Help them see that they also set the translation matrix. **What else will you need to set in this program?** They should note that they will need to set the rotation matrix.

If students have saved their *Move That Line!* programs on their calculators, they could begin the new program by changing the lines of that program. Most graphing calculators allow users to copy lines of an old program into a new program. You might show students how to do this to save them time and avoid tedious work.

With this introduction, have students work in pairs on *Swing That Line!*

Discussing and Debriefing the Activity

Wait until everyone has finished Question 3, if possible. You may want to have students who have finished Question 4 display their programs for the class, so they can run the programs as well as share the written code.

If students were working from the plain-language program for *Move That Line!*, they might get a program like this, in which that same segment is rotated 10 times, through an angle of 15° each time:

Setup program

Let A be the matrix $\begin{bmatrix} 3 & 2 \\ 5 & 1 \end{bmatrix}$

Let B be the matrix $\begin{bmatrix} \cos 15^\circ & \sin 15^\circ \\ -\sin 15^\circ & \cos 15^\circ \end{bmatrix}$

For J from 1 to 11

- Clear the screen
- Draw a line from (a_{11}, a_{12}) to (a_{21}, a_{22})
- Delay
- Replace matrix A by the product $A \cdot B$

End the J loop

It is time for another toot of your trumpet. Students can now check off the second part of item 3—"Rotate the object"—from their unit plan from the discussion introducing *Picture This!* (as emended in the discussion introducing *Goin' Round the Origin*).

Key Questions

What else did you set at the beginning of the *Move That Line!* program, besides the initial coordinates?

What else will you need to set in this program?

Doubles and Differences

Intent

Students develop further trigonometric identities.

Mathematics

Students put their experience in developing formulas for the sine and cosine of a sum of pair of angles to work as they discover more formulas. They formulate expressions for the sine and cosine of a difference of two angles and for the sine and cosine of twice an angle.

Progression

Students develop new trigonometric formulas and check their work by substituting pairs of values for the variables.

Approximate Time

5 minutes for introduction
30 to 35 minutes for activity (at home or in class)
10 to 15 minutes for discussion

Classroom Organization

Individuals, then groups, followed by whole-class discussion

Doing the Activity

In this activity, students will create formulas for the sine and cosine of the difference of two angles and for the sine and cosine of twice an angle. They will likely do so by using formulas already developed, so they will need those formulas in their notes if this activity is assigned as homework.

Use your judgment about whether simply to instruct students to take all their formulas home or to specify which formulas they will need. Here are the required formulas:

- $\sin(A + B) = \sin A \cos B + \cos A \sin B$
- $\cos(A + B) = \cos A \cos B - \sin A \sin B$
- $\cos(-\theta) = \cos\theta$
- $\sin(-\theta) = -\sin\theta$

Discussing and Debriefing the Activity

Ask students to come to consensus in their groups, and have several groups prepare presentations. You might omit having students show their work verifying the formulas.

Then ask several students to make presentations. Here are the formulas:

- $\sin (A - B) = \sin A \cos B - \cos A \sin B$
- $\cos (A - B) = \cos A \cos B + \sin A \sin B$
- $\sin 2A = 2 \sin A \cos A$
- $\cos 2A = \cos^2 A - \sin^2 A$

Supplemental Activity

Half a Sine **(reinforcement)** is a natural follow-up to this activity.

What’s Going On Here?

Intent

Students analyze a program that uses matrix transformations.

Mathematics

This activity applies ideas that students have been working with to a multipoint picture.

Progression

Presented with a program that uses matrices to rotate a picture, students sketch the resulting display on graph paper. During the follow-up discussion, they create and test code for the program.

Approximate Time

30 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

Individuals, followed by whole-class discussion

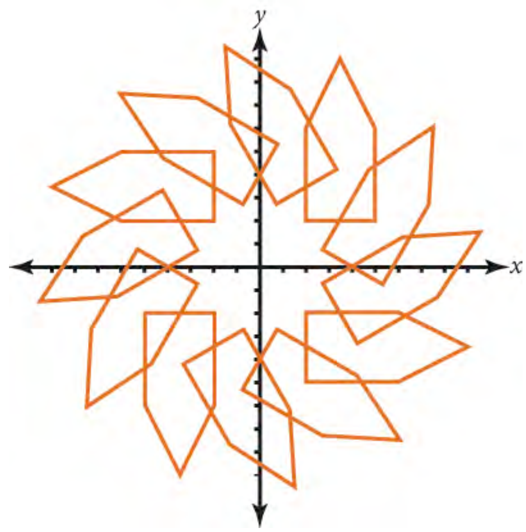
Doing the Activity

No introduction of this activity is required.

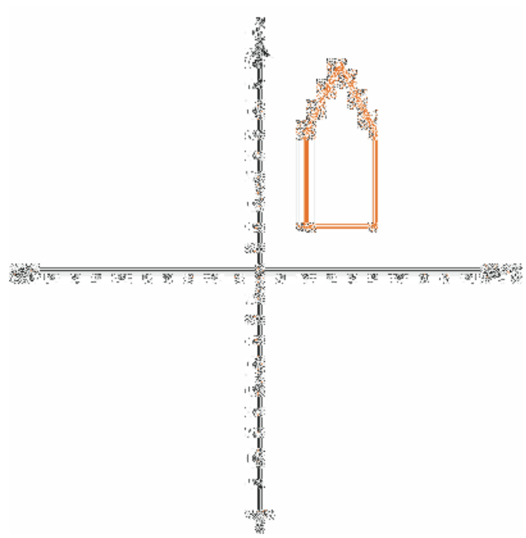
Discussing and Debriefing the Activity

Have various students go step by step through the program, describing what happens and perhaps going through the loop two or three times. You might have the class set up a table to keep track of the changing values for matrix A and loop variable *C* (matrix B doesn’t change) and plot the successive drawings on coordinate axes.

Then have the class create, enter, and run code for the program. The result should look something like this:



The program draws an initial "house" figure, as shown in the next diagram. Then it rotates that figure around the origin in 30° increments, creating 12 separate versions of the picture.



Be sure students see how the house is turning around the origin. For example, the distance from each corner of the house to the origin remains constant throughout the process, and the corner closest to the origin stays closest.

Ask students how the program could be changed to create more of an illusion of motion. **How could the program be modified to show only one picture at a time?** They should see that they could add a delay loop and an instruction to clear the screen within the existing For/End loop. You might suggest they work on this variation in their spare time.

You might also ask, **What causes the figure to make one complete circle with its motion? How would the figure need to be altered if the angle for each rotation were some value other than 30°?**

Key Questions

How could the program be modified to show only one picture at a time?
What causes the figure to make one complete circle with its motion? How would the figure need to be altered if the angle for each rotation were some value other than 30°?

Projecting Pictures

Intent

In these activities, students develop the mathematics behind geometric projections.

Mathematics

Students begin developing the formulas and processes they will need to perform projections by adapting the midpoint formula to find a point an arbitrary fraction of the distance between two points, first in two dimensions and then in three dimensions. After examining a physical model of the projection process, they apply that adaptation to calculate the coordinates of a point projected onto a screen, given the coordinates of the viewpoint and the equation of the plane of the screen.

Although it is not needed for the unit problem, students also look at geometric reflections and how to reflect a point using matrices.

Progression

These activities open with the teacher-led discussion and demonstration *From Three Dimensions to Two.* The teacher introduces the idea of a projection onto a plane, discusses the need to draw three-dimensional objects in two-dimensional space, and demonstrates how to use a line of sight to make a projection.

Before students begin working on the activities dealing with projection, they encounter a partner project, *An Animated POW*, that will require fairly consistent work between now and the end of the unit. In the project, students write a calculator program to create an original animation. After putting some thought into what they want their program to accomplish, toward the end of these activities they outline how the program will do that in *An Animated Outline*.

Students develop and practice working with a generalization of the two-dimensional midpoint formula to arbitrary fractions in *"A Snack in the Middle" Revisited*, *Fractional Snacks*, *More Walking for Clyde*, *Monorail Delivery*, and *Where's Madie?* After a brief review of the three-dimensional coordinate system in *A Return to the Third Dimension*, the midpoint generalization is carried into three dimensions in *And Fred Brings the Lunch*, *Where's Bonita?*, and *Lunch in the Window*.

In *Cube on a Screen,* students work with a physical situation, using their line of sight to draw a projection onto a plane, in preparation for applying the mathematics they have developed to express a projection in terms of coordinates in *Find Those Corners!*

Along the way, students look at how to express reflections using matrices, both in two-dimensional situations (*Flipping Points* and *Further Flips*) and 3-space (*Mirrors in Space*).

Spiders and Cubes gives students an opportunity to reflect upon the connections among these activities. *Where Are We Now?* provides focus for a review of the main ideas that have been developed and how they apply to the unit problem.

From Three Dimensions to Two (teacher-led discussion and demonstration)
POW 9: An Animated POW
"A Snack in the Middle" Revisited
Fractional Snacks
More Walking for Clyde
Monorail Delivery
Another Mystery
A Return to the Third Dimension
Where's Madie?
And Fred Brings the Lunch
Flipping Points
Where's Bonita?
Lunch in the Window
Further Flips
Cube on a Screen
Spiders and Cubes
Find Those Corners!
An Animated Outline
Mirrors in Space
Where Are We Now?

From Three Dimensions to Two

Intent

This teacher-led discussion and demonstration introduces students to the focus of these activities: the mathematics behind geometric projections.

Mathematics

The teacher introduces the idea of a projection onto a plane, discusses the need to draw three-dimensional objects in two-dimensional space, and demonstrates how to use a line of sight to make a projection.

Progression

The teacher conducts a discussion and demonstration.

Approximate Time

10 minutes

Classroom Organization

Whole-class discussion

Materials

Sheet of clear acrylic plastic (Plexiglas) or similar material
Cube (at least 2 inches on an edge)

Doing the Activity

It's time for another brief look at the general plan of the unit. Start with the generic animation outline developed in the discussion following *Move That Line!*, which should look something like this:

Setup program
Set the initial coordinates
Start the loop
- Clear the screen
- Draw the next figure
- Delay
- Change the coordinates

End the loop

Ask students, **What might you need to do to write the program for the turning cube?** They might start by looking at the step of the outline, in which they set the initial coordinates, and should see that they need to decide where the cube is located at the beginning of the program.

Tell them they will get to this in a while, and turn their attention to the next nontrivial step: drawing the next figure.

Ask, **Why might drawing the next figure be difficult?** Emphasize that the cube itself is a three-dimensional figure, but the drawing on the calculator screen will have to be a two-dimensional figure. Explain that this process is called *projecting the figure* and that such a two-dimensional drawing is called a **projection.** This task is item 4 of the unit plan: "Create a two-dimensional drawing of a three-dimensional object."

Let students know that this is not an easy task and that they will spend the next week or so on activities to help them understand how to accomplish it. As they work on these activities, they should be trying to figure out how these activities are related to the idea of projecting from three dimensions to two.

Conducting the Demonstration

It will be helpful to give a brief preview demonstration now of an activity that students will do later, *Cube on a Screen*. The activity and this preview use a cube and a sheet of acrylic plastic (or similar firm, clear material on which students can write). You will need three volunteers to carry out the demonstration.

To begin, ask, **How do you draw a three-dimensional object on a two-dimensional screen?**

Have one volunteer hold the sheet of plastic, the second hold the cube behind the plastic, and the third stand in front of the plastic. Then have the first volunteer point to one of the vertices of the cube, and ask the third volunteer to determine about where on the plastic he or she would draw that vertex, based on how it appears as viewed through the plastic.

The goal here is to suggest that the line from the viewer's eye to the vertex determines the location on the plastic where the point should be drawn. The class's challenge will be to figure out how to express this location on the plastic in terms of the vertex's coordinates and the viewer's eye. However, students need to do considerable work on related ideas before they address this issue directly.

Key Questions

What might you need to do to write the program for the turning cube?
Why might drawing the next figure be difficult?
How do you draw a three-dimensional object on a two-dimensional screen?

POW 9: An Animated POW

Intent

Students write an original animation program.

Mathematics

For this activity, students work in pairs to create animation programs for the calculator. You may want to set some guidelines for your class concerning just how much of the mathematics they have learned you expect to see in their programs. You may want to require that the programs use matrices, use a rotation, or possibly even use a projection (although that last requirement might create difficulties with the timeline for completing the project).

Progression

The POW describes a very open-ended task: work with a partner to create an interesting animation program. The teacher clarifies the expectations for the activity, both in terms of the mathematics employed and the grading criteria, and lays out a schedule for its completion.

Students will probably need about two weeks to complete this project. Within the first few days, they select their partners and develop descriptions of what they intend their programs to do. In *An Animated Outline,* students finish the outlines of their programs.

Students will need ample time in which to meet with their partners outside of class to work on their programs, with at least one class period set aside for working together in class. Each pair of students will eventually make a presentation to the class and hand in their written program.

Approximate Time

15 to 20 minutes for introduction
3 or more hours for program development (at home; time will vary dramatically depending upon the complexity of the animation attempted)
40 to 50 minutes for partners to work together in class
80 to 95 minutes for presentations

Classroom Organization

Pairs, followed by a presentation by each pair

Materials

Calculator manuals (1 per pair of students)

Doing the Activity

Remind students that at the end of the unit, they will write an animation program. *POW 9: An Animated POW* describes this animation project. Students will work in pairs, but everyone needs to turn in an overall outline for the program, a more detailed plain-language version of the program, and the actual code. Take time now to clarify again what level of detail you expect for each of these components.

Options for the POW

The POW asks students to write a calculator program. However, you might allow them other options such as writing a computer program or doing some animation using cutouts and a video camera.

As written, the POW does not specify the mathematical expectations for the project. Feedback from teachers suggests that students may get a great deal out of the activity even if they do not make significant use of the unit's new mathematical ideas. If you want to have students apply some of the mathematical ideas of the unit, you might give them specific guidelines, such as requiring them to use a matrix or asking them to use at least one mathematical idea and report on it in their presentations.

It is probably a good idea to explain your grading criteria. Here are some possible items to include:

- Creativity
- Aesthetics of the visual display
- Elegance of the program
- Use of mathematics

Schedule for the POW

You might give students a schedule for completing the various stages of the POW. The sample schedule below may need to be modified depending upon the length of your class period, the days your class meets, the complexity of the programs attempted, and students' ability to meet outside of class time.

- Day 1: Introduce the POW.
- Day 3: Students hand in their partner selections.
- Day 5: Partners hand in their descriptions of what their programs will do.
- Day 10: Partners hand in a written outline of their program (*An Animated Outline* asks students to finish up their outlines).
- Day 14 or 15: Students complete their POWs in class.

- Days 15 and 16: Partners make presentations and hand in their written programs (see *"An Animated POW" Write-up*).

Discussing and Debriefing the Activity

Plan on about five minutes per presentation, so the presentations may span more than one class period.

Since most of the programs are likely to be too lengthy for the presenters to fully explain the flow of the programming code, some teachers have found it useful to have partners comment on a single programming difficulty they ran into and how they resolved it. These explanations often bring to light some of the most interesting, creative, and challenging aspects of students' programs, which are not otherwise revealed by simply observing the animation.

It might be worthwhile to make a video of students' presentations. You can show the video to other mathematics classes, colleagues, and administrators or use it with parents at an open house.

Peer Reviews of Presentations

You may want to have students write reviews of one another's presentations. If so, remind students of the grading criteria for the POW. You might have students brainstorm and create a rubric for evaluating one another's work, based on those criteria (or their own). For simplicity, consider creating a form students can use in writing their reviews.

Immediately after each presentation, students can work in their groups to discuss and evaluate the quality of the program and the presentation and to write their reports. Pass these reviews on to the presenters after you have read them.

"A Snack in the Middle" Revisited

Intent

Students begin the work of generalizing the midpoint formula for arbitrary fractions.

Mathematics

This activity builds on the situation in *A Snack in the Middle* from the unit *Orchard Hideout,* in which the focus was on developing and at least partially proving the midpoint formula. In this new activity, the desired point is still on a given line segment, but not necessarily halfway between the endpoints. The focus now is on how to modify the midpoint formula as the "fraction of the way" changes. This work will be continued in *Fractional Snacks.*

This activity is the first step toward item 4 of the unit plan—"Create a two-dimensional drawing of a three-dimensional object"—although the connection will probably not be immediately clear to students.

Progression

Students return to the *Orchard Hideout* situation, with Madie and Clyde trying to determine where to place their snack table between them as they work in the orchard. They first find the midpoint of the line segment that would connect the two positions and then examine a more difficult situation where the table is placed one-third of the way from Clyde to Madie. The subsequent discussion reviews the midpoint formula, focusing on proving that the formula works. The class then uses a diagram to illustrate the similar triangles involved in Question 3.

Students generalize their work from this activity in *Fractional Snacks*.

Approximate Time

25 to 30 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

Individuals, followed by whole-class discussion

Materials

Optional: Transparencies of the *"A Snack in the Middle" Revisited* blackline masters.

Doing the Activity

If some students are unfamiliar with the unit *Orchard Hideout,* briefly explain the setup of the orchard, in which trees are planted at lattice points in a coordinate system.

You may want to suggest that students use graph paper for this activity.

Discussing and Debriefing the Activity

The discussion of this and the next couple of activities focuses on issues in the two-dimensional coordinate system in preparation for later work in three dimensions. It is very important that students feel comfortable with the ideas in this simpler setting before they move on to a situation that is often much harder to visualize.

The discussion here focuses on the proof in connection with Question 1 and the computation for Questions 2 and 3.

Question 1

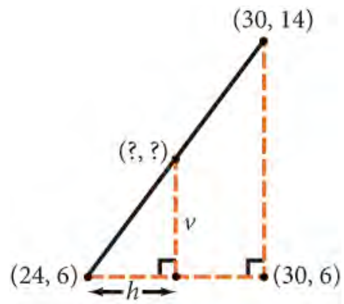
Have a student report on Question 1, and then ask for volunteers who did the computation differently. Ask, **How did you determine your answer?** It will be helpful to see a variety of methods. Here are three likely approaches:

- Students may remember the midpoint formula, which leads to the computation $\left(\frac{24+30}{2}, \frac{6+14}{2}\right)$.
- Students may do something similar on a more intuitive level, saying that 27 is halfway between 24 and 30 and that 10 is halfway between 6 and 14.
- Students may take half the difference in the coordinates in each direction and add these to the values at (24, 6). For example, they may say that since (30, 14) is 6 units to the right of (24, 6), the midpoint should be 3 units to the right of (24, 6). This makes the x-coordinate of the midpoint 24 + 3, or 27. Similarly, the y-coordinate of the midpoint is $6 + \frac{1}{2} \cdot 8$, or 10.

The x- and y-coordinates Work Independently

Before moving on, point out that for all of these methods, students are working with the x- and y-coordinates separately. Ask if anyone can *prove* that any of these methods is correct.

If necessary, suggest students use a diagram like this one, where (?, ?) represents the midpoint. (A large version of this diagram appears on a blackline master.)



The small right triangle is similar to the large one, because they have a common angle at (24, 6) and each has a right angle. But the hypotenuse of the small triangle is, by assumption, half the length of the longer hypotenuse. Therefore, other parts of the small triangle are half the corresponding parts of the large triangle.

Because the vertical leg of the large triangle has length 8 (calculated by 14 – 6), the vertical leg of the small triangle (v) must have length 4. Similarly, the horizontal leg of the small triangle (h) has length 3. Thus, the midpoint is (27, 10). [*Note:* Adding the coordinates of the point at (30, 6) to the diagram will help many students find the lengths 8 and 6.]

The main point here is for students to recognize that the computation rests on ideas about similar triangles. At various times in the work on item 4—"Create a two-dimensional drawing of a three-dimensional object"—you will probably want to ask about the geometric principle behind a computation. Students should be able to identify similarity as the guiding concept.

Save this diagram to refer to during the discussions related to three dimensions.

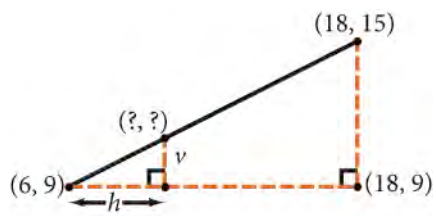
Question 2

Unless students had considerable difficulty with Question 1, simply have another student go over the computation for Question 2, again looking for different methods.

Question 3

Let another student explain how he or she did Question 3. Most likely, he or she will use something analogous to the third of the methods outlined for Question 1, saying that Madie is 12 units to the right of Clyde, so the table should be 4 units to Clyde's right ($\frac{1}{3}$ of 12 units). Similarly, the snack should be 2 units "up" from Clyde ($\frac{1}{3}$ of 6 units).

Ask students to draw a diagram for the situation and then identify similar triangles. The diagram might look like this one. (This diagram also appears on a blackline master.)



The distance from (6, 9) to (?, ?) is $\frac{1}{3}$ of the distance from (6, 9) to (18, 15).

Reasoning with similar triangles, h is $\frac{1}{3}$ of 12 and v is $\frac{1}{3}$ of 6.

Some students may say the table should be at (4, 2), using the values of h and v and forgetting the final step of adding 4 units to Clyde's x-coordinate and 2 units to his y-coordinate. Be sure to identify this step in the computation.

Key Question

How did you determine your answer?

Fractional Snacks

Intent

Students find the point a certain fraction of the way along a line segment.

Mathematics

This activity completes the generalization of the midpoint formula for arbitrary fractions that was begun in *"A Snack in the Middle" Revisited*.

Progression

The activity begins with a couple of questions similar to the third question from *"A Snack in the Middle" Revisited* and then asks students to generalize the computation using *r* as the fraction of the distance. The subsequent discussion brings out that the midpoint formula is a special case of this new formula and suggests that students begin to think about how this work relates to the unit problem.

Approximate Time

35 to 40 minutes

Classroom Organization

Small groups or individuals, followed by whole-class discussion

Doing the Activity

Tell students that in this activity, they will generalize the ideas from *"A Snack in the Middle" Revisited*. (*Note:* The letter *r* is used in Question 3 to suggest the word *ratio*.

Discussing and Debriefing the Activity

Question 1

Ask a student to explain how she or he did Question 1. The method will probably be similar to that used for Question 3 of *"A Snack in the Middle" Revisited*.

Ask, **How is this computation different from that in Question 3 of *"A Snack in the Middle" Revisited*?** This question will help students identify how they are using the fraction $\frac{1}{4}$ from the problem.

You may also want to use this problem to get students to focus on the computation by which they get the two components of the distance from Clyde to Madie. For example, if the presenting student says that Madie is 16 units to the right of Clyde, ask how he or she found that distance. Getting students to express this explicitly as $14 - (-2)$ will help them formulate the general case.

Question 2

Watch out for two changes as students discuss this problem:

- In previous questions, students may have divided the appropriate distances by 3 or 4, rather than multiplying by $\frac{1}{3}$ or $\frac{1}{4}$. They may have some trouble with Question 4 because the fraction involved is not a unit fraction (a fraction with numerator equal to 1). You may need to talk about multiplying by the appropriate fraction rather than dividing.
- In the earlier problems, each of Madie's coordinates was greater than Clyde's corresponding coordinate. Students may have some trouble in Question 2 dealing with the signs of the differences between the coordinates.

Both of these issues deserve attention, especially in preparation for the generalization. Use the discussion of Question 2 to bring out, for instance, that one can always subtract Clyde's coordinates from Madie's, multiply by the fraction, and add the results to Clyde's coordinates. If Madie's coordinates are less than Clyde's, it simply means the differences will be negative.

Question 3

Ask for volunteers on Question 3. It is more difficult than Questions 1 and 2, so students may have worked on it at different levels.

Some students may express the description verbally, along these lines: "Find the distances from Clyde to Madie in each of the *x*- and *y*-directions, take the right fraction of these distances, and add these partial distances to Clyde's position."

Others may formulate the process algebraically. For example, they may give both Clyde's and Madie's initial coordinates symbolically and then give a formula for the coordinates of the table.

Validate both approaches, but work toward getting everyone to see the algebraic approach, building on the concrete examples. Help students focus on two specific elements:

- The general approach of "original coordinates plus a fraction of the distance between them"
- The role of similar triangles, including the fact that the same "fraction of the distance" is seen for each pair of sides of the similar triangles

Students should develop a generalization something like this:

If r is any fraction, then the point that is "r of the way" from (x_1, y_1) to (x_2, y_2) has these coordinates:

$$(x_1 + r(x_2 - x_1), y_1 + r(y_2 - y_1))$$

Post this general rule. You might also have students connect this formula back to their verbal descriptions. Thus, $x_2 - x_1$ and $y_2 - y_1$ represent "finding the distances from Clyde to Madie in each of the x- and y-directions," multiplying by r represents "taking the right fraction of these distances," and the sums $x_1 + r(x_2 - x_1)$ and $y_1 + r(y_2 - y_1)$ represent "adding these partial distances to Clyde's position."

The Midpoint Formula

Regardless of whether students referred explicitly to the midpoint formula in the discussion of Question 1 of *"A Snack in the Middle" Revisited*, ask, **What does the general formula just developed tell you about the case of finding the midpoint?**

One purpose of this question is to have students recognize that the midpoint is the special case $r = \frac{1}{2}$. The other is to give them an opportunity to apply the general formula, substituting $\frac{1}{2}$ for r, and simplify the algebra. That is, they should see that the expression

$$(x_1 + \frac{1}{2}(x_2 - x_1), y_1 + \frac{1}{2}(y_2 - y_1))$$

simplifies to

$$\left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}\right)$$

which is the midpoint formula.

Snacks and Turning Cubes

Tell students they should be thinking about what the snack problems have to do with the problem of programming a turning cube—in other words, they should be considering the question, **How does the midpoint formula relate to the unit question?** This does not need to be discussed now, although students may realize that this formula will help them compute the coordinates of a projected point.

Key Questions

How is this computation different from that in Question 3 of *"A Snack in the Middle" Revisited*?
What does the general formula tell you about the case of finding a midpoint?
How does the midpoint formula relate to the unit question?

More Walking for Clyde

Intent

Students practice finding the point a certain fraction of the way along a line segment.

Mathematics

This activity continues the general idea presented in *"A Snack in the Middle" Revisited* and *Fractional Snacks*.

Progression

Students explore several problems where they must find a point a certain fraction of the way between two given points. The discussion brings out a variety of approaches.

Approximate Time

20 to 30 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

Individuals, followed by whole-class discussion

Doing the Activity

This activity requires no introduction.

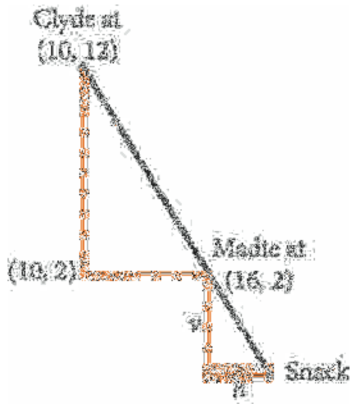
Discussing and Debriefing the Activity

The problems in this activity are quite similar to those in *"A Snack in the Middle" Revisited*. There are several approaches students might use; elicit a variety of ideas.

If students work with the numbers as given (that is, using $\frac{1}{2}$ in Question 1 and 2 in Question 2), they should realize that the *x*- and *y*-components must be added to Madie's coordinates instead of to Clyde's.

For example, in Question 1, using a diagram like the one shown here, students can see that h is half of $16 - 10$ and that v is half of $2 - 12$. These values are added to

Madie's coordinates to show that the table belongs at (16 + 3, 2 + (−5)), or (19, −3). (*Note:* Students may prefer to think of v as 5 rather than as −5 and to subtract 5 from 2 rather than add −5 to 2.)



An alternate approach is to say that the table's distance from Clyde is $1\frac{1}{2}$ times the distance from Clyde to Madie and to use the number $1\frac{1}{2}$ as the value for r in the formula developed in *Fractional Snacks*.

Use your judgment about whether to develop a general formula for the "beyond Madie" situation. It will not be needed in this unit.

Monorail Delivery

Intent

Students find the point where a given vertical line meets a given line segment.

Mathematics

In this activity, a new twist is given to the problem of finding a fractional distance from two points. Now rather than giving the fraction, the problems give the equation of a vertical line. Students must find the intersection of this line with the segment that joins the two given points.

Progression

Students analyze two problems with specific numbers and then generalize their process using variables in place of the numbers for two points. The subsequent discussion focuses on how to use the known x-coordinate to find the desired ratio.

Approximate Time

40 to 50 minutes

Classroom Organization

Small groups or individuals, preceded and followed by whole-class discussion

Doing the Activity

This activity continues the situation of determining where Clyde and Madie should put their snack. This time, students must use the information in the problem to find the fraction that corresponds to the number r in the formula from *Fractional Snacks* (this fraction was given to them in the previous snack problems). When students develop their programs to turn the cube, finding a similar fraction will be an important element of the process.

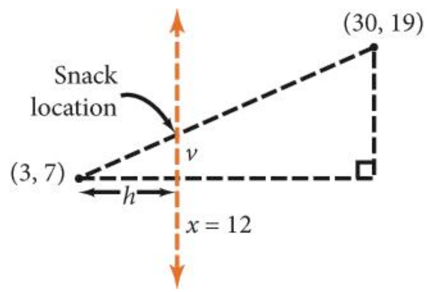
Have students read the activity, individually or as a class. Then ask, **How is this activity different from *More Walking for Clyde*?** If necessary, also ask, **What information do you have, and what don't you have?** Help students as needed to see that in the previous snack problems, they knew what fraction of the way the snack was located on the line segment between Clyde and Madie.

With that clarification, have groups begin work. If they get stuck, ask, **How might you find that fraction in these problems?** If necessary, ask where else the fraction appeared in the previous problems (aside from its explicit mention). Students might remember that the fraction also shows up in a comparison of the vertical (and also the horizontal) sides of a small triangle to those of a similar large triangle.

In these problems, students can find the ratio of horizontal sides directly because the problem says, in effect, that the x-coordinate of the snack is 12.

Discussing and Debriefing the Activity

Have students from various groups present Questions 1 and 2. For Question 1, students might use a diagram like this one:



In this diagram, the distance h can be found directly as 12 – 3, because the monorail is along the line $x = 12$. The key step is realizing that the fraction called r in the earlier formula is given by the ratio $\frac{12-3}{30-3}$, or $\frac{1}{3}$. This fraction can then be used to find v (which is one-third of 19 – 7), and then the value of v can be added to 7 to get the y-coordinate of the snack's location. Thus, for Question 1, the snack should be dropped off at the point (12, 11). Question 2 is similar.

Note: Some students might want to get the equation of the line connecting the two points and use that to find the intersection with the monorail. If so, acknowledge that it is a good idea in this case, but explain that the method would not generalize well to the three-dimensional issues awaiting them in the unit problem.

Now ask for a volunteer to present ideas on how to create a general formula (still using $x = 12$ as the equation of the monorail). As needed, help the class work from the specific examples to get that in general, the fraction can be expressed as $\frac{12-a}{c-a}$ (if distance is measured from b). If this fraction is labeled r, the position of the snack can be expressed as

$$(12,\ b + r(d - b))$$

Optional: A Further Generalization

In preparation for later work in this unit, you may want to have students generalize this expression further, using an arbitrary line $x = k$ to represent the monorail. In this case, r is the fraction $\frac{k-a}{c-a}$, and the position of the snack can be expressed as $(k, b + r(d - b))$.

Key Questions

How is this activity different from *More Walking for Clyde*? What information do you have, and what don't you have?

How might you find the fraction in these problems?

Another Mystery

Intent

Students analyze a program combining rotation and translation using matrices.

Mathematics

This is the first program students have seen that combines a rotation with a translation. This program is fairly easy to follow conceptually, but working out the details of exactly what gets drawn is not quite as easy.

Progression

Presented with a plain-language program, students draw what the calculator display would show after running the program and then translate the program into programming code. In the subsequent discussion, they enter the code and test their programs.

Approximate Time

30 minutes for activity (at home or in class)
15 minutes for discussion

Classroom Organization

Individuals, then groups, followed by whole-class discussion

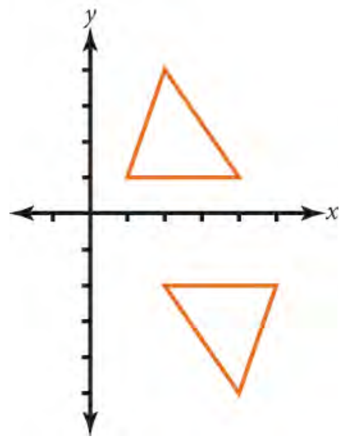
Doing the Activity

Students work on the activity independently.

Discussing and Debriefing the Activity

Have students compare notes on the program in their groups to see if they can come to a consensus about what it does. Then have them enter the code they created and run the program to see what actually happens. If their predictions were incorrect, let them take some time in their groups to try to figure out what went wrong with their analysis.

With appropriate window settings, the screen will look something like this:



Students may be surprised that there are only two triangles, because the loop goes from W = 1 to W= 4. It turns out that when the triangle is rotated and translated a second time, it returns to its original position.

Students may realize that the program involves a combination of a translation and a rotation and may also see that the net effect appears to be a rotation of the triangle around the point (3, –0.5). If this comes up, explain that any combination of a rotation and a translation is, in fact, another rotation, but not necessarily a rotation around the origin. Students may find this observation useful for *POW 9: An Animated POW.*

A Return to the Third Dimension

Intent

Students review the three-dimensional coordinate system.

Mathematics

This activity helps students recall what they know about three-dimension graphing, including the coordinate system and equations of planes in that system.

Progression

The teacher introduces the topic by setting up a three-dimensional coordinate system and having students locate various points in that system. After students (optionally) build models of the coordinate system, the teacher reviews that the graph of a one-variable linear equation is a plane parallel to a coordinate plane and challenges students to visualize the polyhedron formed by a given set of vertices. The remainder of the activity reviews basics about the three-dimensional coordinate system. No subsequent class discussion is necessary.

Approximate Time

35 to 40 minutes

Classroom Organization

Small groups or individuals, preceded by whole-class discussion

Materials

Large (5″ × 8″ or larger) index cards (3 per group)
Yarn and a pair of scissors for each group
Optional: Transparency of *A Return to the Third Dimension* blackline master
Optional: String for making a large-scale, three-dimensional coordinate system

Doing the Activity

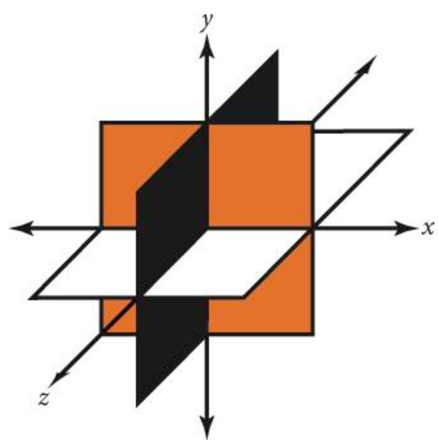
Students often find it helpful to use the classroom as a model of the three-dimensional coordinate system, so you may want to construct such a model before students begin the activity.

There are several ways to set up the classroom coordinate system (assuming your classroom is a typical "box" shape). For consistency with the opening-day demonstration, use the board as the standard *xy*-plane, with the positive *z*-axis coming out perpendicular to that plane.

Alternative Approach

The following alternative approach allows you to consider both positive and negative coordinates within the classroom. First, choose a point in midair somewhere in the middle of the classroom. This point will represent the origin. Then tape three pieces of string across the room, through this point, to form straight lines in each of the three perpendicular directions (floor to ceiling, front to back, left to right). These pieces of string will represent the axes.

Label the strings as the *x*-axis, *y*-axis, and *z*-axis, and assign one end of each axis as the positive direction. Have the *x*-axis connect the side walls, the *y*-axis connect the floor and ceiling, and the *z*-axis connect the front and rear walls. That way, the *x*-axis and *y*-axis are parallel to their usual chalkboard positions in the two-dimensional coordinate system. Viewed from the back of the room, the situation looks roughly like this diagram, with the coordinate planes shown for clarity. (This diagram is included on a blackline master.)



Note: In this setup, if students at the front of the room turn around to look at the *x*-axis, its positive direction will be to their left. This may be confusing, so be sure to identify the positive direction for each axis clearly.

Some teachers prefer to set up the axes with the origin in a corner on the floor of the classroom. In that approach, points in the classroom have positive coordinates.

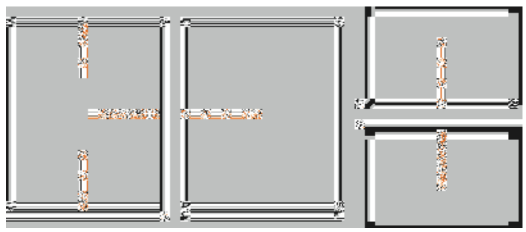
Reviewing the Coordinate System

Once you establish the setup of the axes, continue the review by having different students first locate various ordered triples as points in the room and then deduce the coordinates of some given points from their locations.

Building a Small Model

Question 1 suggests that students use a small-scale model of the coordinate system to understand the situation. Many students find such a model helpful and can build one using large index cards as coordinate planes. Have each individual or group build a model now and save it for use over the rest of the unit.

To do this, have each group start with two full-size index cards and two half cards, with slits cut as indicated by the dashed lines.



Students first fit the two larger cards together along the slits shown here as horizontal, keeping the first card vertical and turning the second to lie horizontal. They then fit the two smaller cards so their slits match those at the top and bottom of the first card, turning the smaller cards to be perpendicular to both of the larger cards.

Once students have constructed these models, have them practice locating points within this system.

Graphs of One-Variable Linear Equations

Ask students what the graph is in this three-dimensional system for a particular equation. **What is the graph of the equation $z = 5$ in this coordinate system?** If needed, ask them to give you some ordered triples that satisfy the equation and to locate those points. They should see that they get a plane that can be described either as perpendicular to the *z*-axis or as parallel to the *xy*-plane.

Then turn the process around, starting with a description of a plane parallel to one of the coordinate planes and asking for its equation. You might ask students to name some points on the plane and then to think about what those points have in common.

Identifying a Polyhedron

As a final element in the review, ask students to imagine a polyhedron with four vertices that is placed in this coordinate system so that the vertices are located at (0, 0, 0), (4, 0, 0), (2, 0, 4), and (2, 4, 2). Give students a few minutes to work within their groups on visualizing this polyhedron.

Then ask for a description. **What does this polyhedron look like?** Students should be able to articulate that this is a tetrahedron, in the first octant, and that its base is a triangle in the xz-plane. If students do not know the term *tetrahedron,* introduce it here.

Elicit such details as that one side of the base is along the x-axis, that the base is an isosceles triangle, and that the tetrahedron is 4 units "high" because vertex (2, 4, 2) at the "top" is 4 units directly above (2, 0, 2), which is a point of the base.

If more work along this line seems needed, ask similar questions requiring students to visualize objects in three dimensions.

A Return to the Third Dimension

After this introduction, turn students loose on the activity, which may not need a whole-class discussion. Questions 2 and 3 should be considered optional and are intended for groups that finish Question 1 quickly.

Key Questions

What is the graph of the equation $z = 5$ in this coordinate system?
What does this polyhedron look like?

Where's Madie?

Intent

This activity will give students more experience with the monorail idea.

Mathematics

Students continue working in two dimensions, finding points a certain fraction of the way along a line segment. This idea will be extended to a three-dimensional situation in subsequent activities.

Progression

Given Clyde's location, the location of the snack, and the fraction of the distance, students find Madie's location.

Approximate Time

25 to 30 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

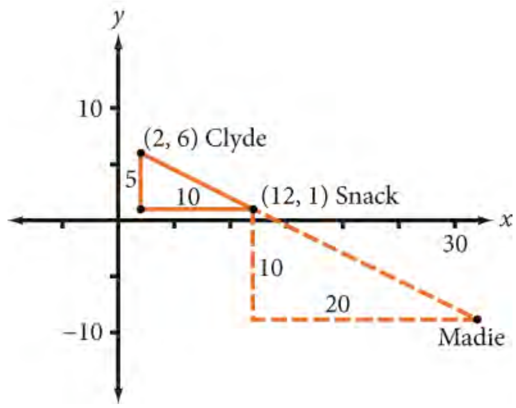
Individuals, followed by whole-class discussion

Doing the Activity

Students can begin working without an introduction.

Discussing and Debriefing the Activity

Ask a volunteer to report on both Questions 1 and 2. For Question 1, the presenter might draw a right triangle using Clyde's and the snack's positions as two vertices (as shown below), extend the hypotenuse so the new portion is twice the length of the original hypotenuse, and create a new right triangle with the new portion as the hypotenuse.



Elicit a clear explanation of how to use such a diagram to see that Madie's tree is at (32, –9).

Then have the same student present Question 2. As he or she gives the general description, have the class apply it to the situation in Question 1 to verify that it matches what the student did in Question 1.

Next, ask for volunteers to describe different methods for answering Question 1 and then to share the instructions they developed in Question 2.

Finally, ask for volunteers to present a general formula. Students should get something equivalent to $(12 + 2(12 - a), c + 2(c - b))$. Have students verify that if $a = 2$, $b = 6$, and $c = 1$ (as in Question 1), they get the same result they found for Question 1. You may also want to have them test out the instructions developed for Question 2 to see if they yield the same formula (or an equivalent one).

And Fred Brings the Lunch

Intent

Working toward a formulation for projecting a representation of a three-dimensional object onto a two-dimensional screen, students now find points a certain fraction of the way along a line segment in three dimensions.

Mathematics

Students will adapt to three dimensions the generalizations of the midpoint formula that they developed previously.

Progression

Students work in groups or individually on an activity similar to the orchard snack problems, but in three dimensions. In the subsequent discussion, they justify their work using similar triangles.

Approximate Time

70 to 80 minutes

Classroom Organization

Small groups or individuals, followed by whole-class discussion

Doing the Activity

Tell students that this activity presents a problem similar to the orchard snack problems, but in three dimensions. You may want to suggest they build a physical model of the situation using cardboard or visualize the situation in terms of their classroom coordinate system.

Ask each group to prepare a presentation on one of the questions.

Discussing and Debriefing the Activity

The individual problems in this activity are all three-dimensional analogs of the orchard snack problems. Begin by having various students or groups present Questions 1 to 4.

For Questions 1 and 2, students can adapt the midpoint formula to three dimensions to get the lunch locations as $\left(\frac{0+11}{2}, \frac{0+13}{2}, \frac{0+24}{2}\right)$ and $\left(\frac{2+9}{2}, \frac{1+12}{2}, \frac{4+20}{2}\right)$, respectively.

For Questions 3 and 4, students will need to adapt to three dimensions the generalization of the midpoint formula they developed in *Fractional Snacks* (or reconstruct the reasoning used there).

For example, in Question 3, Bonita is 7 units to Charlotte's right, 11 units above her, and 16 units forward from her. Because the lunch is one-third of the way from Charlotte to Bonita, it should be placed $\frac{7}{3}$ units to Charlotte's right, $\frac{11}{3}$ units above her, and $\frac{16}{3}$ units forward from her, or at $(4\frac{1}{3}, 4\frac{2}{3}, 9\frac{1}{3})$.

Similarly, in Question 4, the lunch should be located at the point $(2 + \frac{2}{5} \cdot 7, 1 + \frac{2}{5} \cdot 11, 4 + \frac{2}{5} \cdot 16)$, or $(4\frac{4}{5}, 5\frac{2}{5}, 10\frac{2}{5})$.

Ask for a volunteer for Question 5, in which r represents some fraction of the way from (x_1, y_1, z_1) to (x_2, y_2, z_2). Students should see that the lunch should be located at

$$(x_1 + r(x_2 - x_1), y_1 + r(y_2 - y_1), z_1 + r(z_2 - z_1))$$

and recognize this as a three-dimensional version of the formula developed for *Fractional Snacks*.

Ask students, **How do you know that the reasoning you used in the two-dimensional case works here?** If needed, ask them to think back to the two-dimensional case and redevelop the reasoning used there. They should see that the argument was based on similar triangles.

Similar triangles also work for the three-dimensional case, but the reasoning involves a two-stage process and is more difficult to visualize. You can be satisfied if students simply recognize that similar triangles are the necessary concept here, without getting into all of the details.

Students may also argue that "of course" you can simply add a third coordinate to the formula. If so, point out the difference between something seeming reasonable and something being proved.

Key Question

How do you know that the reasoning you used in the two-dimensional case works here?

Flipping Points

Intent

This activity will give students more experience working with coordinate geometry. This material is not needed for solving the unit problem.

Mathematics

Students express **reflections** in two dimensions using coordinates and matrices.

Progression

Flipping Points illustrates the reflection of a triangle, using the y-axis as the line of reflection. Students give the coordinates of the vertices of the triangle and its reflection, express the coordinates of the reflected points in terms of the coordinates of the original points, and then find a way to represent this transformation using a matrix.

Approximate Time

25 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

Individuals, followed by whole-class discussion

Doing the Activity

No introduction to this activity is necessary.

Discussing and Debriefing the Activity

Let students report on Questions 1 and 2. These should be fairly straightforward, with students seeing that the reflection of the point (a, b) is the point ($-a$, b). You may want to bring out explicitly that this is true even if a or b is already negative.

Ask for a volunteer for Question 3. If no one was able to answer this question, tell students to try matrix multiplication. If they need a more explicit hint, tell them to look for a matrix [B] such that $\begin{bmatrix} a & b \end{bmatrix} \cdot \begin{bmatrix} \text{B} \end{bmatrix} = \begin{bmatrix} -a & b \end{bmatrix}$. If necessary, set up [B] as $\begin{bmatrix} r & s \\ t & u \end{bmatrix}$. Students should see that they want [B] = $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$.

Supplemental Activity

***The General Isometry* (extension)** involves the process of combining translations, rotations, and reflections. It asks students to show that every isometry in two dimensions can be obtained as a combination of a translation, a rotation, and, if needed, a reflection.

Where's Bonita?

Intent

Students continue to work toward the projection of a three-dimensional object onto a plane.

Mathematics

This activity is a variation on *And Fred Brings the Lunch* and a three-dimensional version of *Where's Madie?* Students are given one end of a line segment, a point in the interior of the segment, and the fraction of the length of the segment that is represented by the distance between the two given points. They must find the coordinates of the other end of the segment.

Progression

This activity can be assigned even if students have not yet found the general formula from *And Fred Brings the Lunch.*

Approximate Time

30 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

Individuals, then groups, followed by whole-class discussion

Doing the Activity

This activity needs no introduction.

Discussing and Debriefing the Activity

Give groups a few minutes to compare ideas, and then let students report on each problem, explaining how they found their answers.

For instance, in Question 1, Charlotte goes 4 units to the right, 5 units up, and 7 units forward to get from her position, (1, 3, 2), to Fred's position, (5, 8, 9). Because this point is halfway to Bonita, Bonita must be 4 units to the right, 5 units up, and 7 units forward from (5, 8, 9), which puts her at (9, 13, 16). Students might explain this by saying that to get from Charlotte to Fred, you "add" (4, 5, 7), so you add the same amount to get from Fred to Bonita.

For Question 2, students need to see that if Fred is one-third of the way from Charlotte to Bonita, then the distance from Fred to Bonita is twice as far (in each of the three directions) as that from Charlotte to Fred.

There is no need to develop a general formula for this process.

In Question 3, if the fact that the fraction is not a unit fraction caused difficulties, give a hint such as, **What is the *x*-coordinate of the point that is one-fifth of the way from Charlotte to Bonita?**

Lunch in the Window

Intent

Students find the point where a given plane intersects a given line.

Mathematics

In this activity, students will essentially accomplish the task of projecting from three dimensions to two dimensions, although they may not yet recognize that they have done so.

Progression

Students find the point where a given vertical plane meets a line segment in 3-space. The teacher brings the class together for discussion of Question 1, and then groups return to the activity. The subsequent discussion points out the similarity of this question to the monorail situation, and the class posts a general formula for the situation.

Approximate Time

40 minutes

Classroom Organization

Small groups, interspersed with whole-class discussion

Doing the Activity

Have students read the introduction to *Lunch in the Window* and work in groups on Question 1. Then bring the class together to discuss that part of the activity.

Begin the discussion by bringing out the similarity between this activity and *Monorail Delivery,* perhaps first asking what previous activity *Lunch in the Window* resembles. **Which previous activity is this like? In what way?** The class should see that the plane of the window is analogous to the line along which the monorail runs.

Then turn to Question 1. Students should recognize that all points in the plane containing the window are 14 feet to the right of the origin, so this is the plane $x = 14$.

Next, ask, **What does the equation of the plane tell about the coordinates of the point where the lunch should be located?** Students should be able to state that the lunch position must be a point whose *x*-coordinate is 14.

Have groups resume work on the activity. If they are stuck on Question 2, ask, **What fraction of the distance (from Charlotte to Bonita) must Fred Jr. travel in the *x*-direction?**

When all groups have finished Question 3, begin the discussion. If no group has finished Question 5, let groups return to it after discussing the earlier problems.

Discussing and Debriefing the Activity

For Question 2, many students will probably have noticed that 14 (Fred Jr.'s *x*-coordinate) is halfway between 4 and 24 (Charlotte's and Bonita's *x*-coordinates), so the window must be halfway between them in terms of left/right position. The reasoning used in *And Fred Brings the Lunch* shows that the lunch spot must also be halfway in the other two directions. In other words, the desired lunch spot is exactly at the midpoint between Charlotte and Bonita, (14, 11, 5).

You may want to bring out again the analogy between this problem and *Monorail Delivery*. Students should recognize that in both problems, they are given one of the coordinates of the desired point and that the key step is using that coordinate to find the value of *r*.

In Question 3, they can use similar reasoning, although the arithmetic is less obvious. Here, the spiders' *x*-coordinates are 10 and 26. Students with good number sense will have seen that the desired point in the window must be one-fourth of the way from Charlotte to Bonita, because 14 is one-fourth of the way from 10 (Charlotte's *x*-coordinate) to 26 (Bonita's *x*-coordinate). And because the desired point is one-fourth of the way from Charlotte to Bonita in the *x*-direction, it must be one-fourth of the way in all directions.

Ask, **How could you find the value $r = \frac{1}{4}$ computationally?** Students should be able to identify this as the ratio $\frac{14-10}{26-10}$.

It may help to introduce the term *x-distance* (or *x-component of the distance*) to describe the difference in *x*-coordinates between two points. Using this term, students might express the computation like this: "Because the lunch spot is one-fourth of the way from Charlotte to Bonita in terms of *x*-distance, it should also be one-fourth of the way from Charlotte to Bonita in terms of *y*- and *z*-distance."

Once they have found *r,* students can either use the formula they developed in *And Fred Brings the Lunch*

$$(x_1 + r(x_2 - x_1), y_1 + r(y_2 - y_1), z_1 + r(z_2 - z_1))$$

substituting $\frac{1}{4}$ for r, or work more intuitively.

If time is short, you do not need to discuss Question 4. The principle here is the same, but the numbers are a bit messier.

Question 5: A General Rule for Fred Jr.

Next, turn to the generalization called for in Question 5. If no group finished this question, let groups work on it now. Students should come up with something equivalent to $\frac{14 - x_1}{x_2 - x_1}$ as the value of r, that is, as the ratio between the distance from Charlotte to the lunch spot and the distance from Charlotte to Bonita.

The expression $r(y_2 - y_1)$ says how far above Charlotte the lunch spot is (that is, the y-distance from Charlotte to lunch), so the y-coordinate of the lunch spot is $y_1 + \left(\frac{14 - x_1}{x_2 - x_1}\right)(y_2 - y_1)$. Similarly, $r(z_2 - z_1)$ says how far forward from Charlotte the lunch spot is, so the z-coordinate of the lunch spot is $z_1 + \left(\frac{14 - x_1}{x_2 - x_1}\right)(z_2 - z_1)$.

Help students confirm this reasoning by asking what the algebra pattern in these formulas would give as the x-coordinate of the lunch spot. Students should see that the analogous expression is $x_1 + \left(\frac{14 - x_1}{x_2 - x_1}\right)(x_2 - x_1)$, which simplifies to $x_1 + (14 - x_1)$, which is equal to 14. They should recognize that this is what it ought to be, because the lunch spot is in the plane containing the window, whose equation is $x = 14$.

Finally, have students generalize to the case where the window is in the plane $x = k$. Then post a summary of the formulas, which might look like this:

> **The line connecting the points (x_1, y_1, z_1) and (x_2, y_2, z_2) meets the plane $x = k$ at the point**
>
> $$\boldsymbol{(k, y_1 + r(y_2 - y_1), z_1 + r(z_2 - z_1))}$$
>
> **where r is the ratio $\frac{k - x_1}{x_2 - x_1}$.**

Key Questions

Which previous activity is this like? In what way?
What does the equation of the plane tell you about the coordinates of the point where lunch should be located?
What fraction of the distance (from Charlotte to Bonita) must Fred Jr. travel in the *x*-direction?

How could you find the value of $r = \frac{1}{4}$ computationally?

Further Flips

Intent

Students express reflections using matrices.

Mathematics

This activity continues the ideas of *Flipping Points*. Students work with reflections in the plane through other lines than that in the earlier activity and express the reflections using matrices.

Progression

Students consider reflections through the lines $y = x$ and $x = 6$. The follow-up discussion brings out that not every reflection can be achieved through matrix multiplication.

Approximate Time

25 to 30 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

Individuals, followed by whole-class discussion

Doing the Activity

No introduction to this activity is needed.

Discussing and Debriefing the Activity

The initial situation in Questions 1 to 3, involving reflection through the line $y = x$, should be fairly straightforward. Students should see (perhaps using the vertices of the triangle as a model) that the image of (a, b) is (b, a).

Based on their experience with *Flipping Points,* students will probably know to look for a matrix [M] such that $\begin{bmatrix} a & b \end{bmatrix} \cdot \begin{bmatrix} \text{M} \end{bmatrix} = \begin{bmatrix} b & a \end{bmatrix}$. A little experimentation should show them that they want [M] = $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

For Question 4 (reflection of the original triangle through the line $x = 6$), students may have some difficulty. With the use of a diagram, they should be able to get the images of the individual vertices:

$$(1, 2) \rightarrow (11, 2)$$
$$(2, 2) \rightarrow (10, 2)$$
$$(1, 4) \rightarrow (11, 4)$$

If needed, suggest they try other points and make an In-Out table, focusing in particular on the x-coordinates. They should see that the image of (a, b) is $(12 - a, b)$. But they may struggle to reach the conclusion that this transformation can't be achieved by matrix multiplication. Either leave this as an open question or suggest they look at *Another Mystery* for ideas.

The transformation in *Another Mystery* was not a reflection, but it did illustrate the idea of combining basic transformations. Specifically, it combined a rotation around the origin with a translation. The reflection in Question 4 can be achieved as a combination of a reflection through a line through the origin and a translation. It can be represented using a combination of matrix multiplication and matrix addition.

Cube on a Screen

Intent

Students use a physical situation to draw the projection of a cube onto a plane.

Mathematics

In this activity, students use their line of sight to draw the projection of a cube onto a plane. In animated films such as *Star Wars*, objects zoom by as we look out the window of a moving spaceship. To make this look realistic, the animator must change the object's shape to reflect the fact that the viewpoint is changing or that the distance between the viewer and the object is changing. The purpose of having student pairs do several drawings in this activity is for them to see that the drawings will look different.

Progression

Students draw the projection of a cube onto a transparent screen. The teacher first demonstrates the task for students. Students then work in pairs and observe how the projection changes as the viewer moves and as the cube moves closer to the screen.

The class comes together for a discussion of Part I before working on Part II, where pairs observe how the projection changes as the cube is rotated. The subsequent discussion emphasizes that the drawings are actually different and not simply translations or rotations of one another. The discussion concludes with consideration of how the activity is related to the unit problem and with a suggestion that students begin thinking about how to represent projection in terms of coordinates. In *Find Those Corners!*, students will apply the concepts from *Lunch in the Window* to the unit problem, specifically to item 4: "Create a two-dimensional drawing of a three-dimensional object."

Approximate Time

40 minutes

Classroom Organization

Pairs, interspersed with whole-class discussion

Materials

Sheets of clear acrylic plastic (Plexiglas) or similar material (1 sheet for each pair of students)
Pens of three colors for writing on the plastic (3 pens for each pair of students)
Cubes (1 per pair, at least 2 inches on an edge, preferably made by connecting smaller cubes with different colors at the corners)

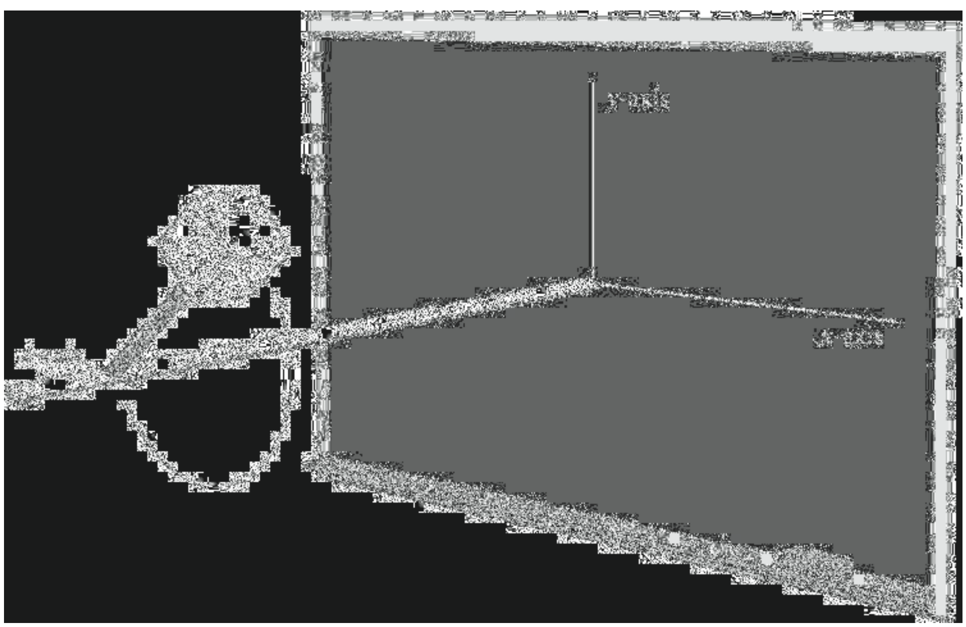
Doing the Activity

Cube on a Screen is a concrete introduction to the more computational activity in *Find Those Corners!* It gives students a chance to carry out in detail the teacher demonstration in *From Three Dimensions to Two.*

Reviewing the Motion of the Cube

Give each pair of students a large colored cube, a sheet of plastic, and three pens of different colors. If possible, have students create this large cube from small colored cubes so that there are different colors at the corners. The colors will help students keep track of their work.

Ask students to summarize the central unit problem, which is to show a rotating cube on a calculator screen. You may want to review the roles and positions of the three axes: the *x*- and *y*-axes are the horizontal and vertical axes in the plane of the wall or board, and the *z*-axis is perpendicular to this *xy*-plane. The cube is going around the *z*-axis and makes one complete spin as it goes around.



Have students work in pairs to act out the motion using their cubes. One can hold the plastic, which represents the screen, while the other turns the cube from behind the screen.

Beginning the Activity

Have students read through the instructions as a class, or explain the sequence of tasks to them. Because the directions are complex, you might demonstrate what to do, perhaps reviewing the demonstration in *From Three Dimensions to Two*. In particular, demonstrate what is meant in Question 6 by the statement "partner B moves the cube through a partial rotation (such as 45° or 90°) around the *z*-axis."

Once students begin, circulate to make sure they have the right picture of what is happening. Discuss Part I as soon as all pairs have completed it, and then let them continue with Part II.

If partner B moves only slightly between Questions 1 and 2, or if partner A moves the cube only slightly between Questions 2 and 3, the differences may not be readily apparent in the tracing. If this point is not clear, you may want to have students make a more dramatic change in position so they do see different faces.

Spiders and Cubes does not depend on the discussion of this activity, so if it is necessary to split *Cube on a Screen* between two class periods, *Spiders and Cubes* can be assigned as homework on the first of the two days.

Discussing and Debriefing the Activity

Part I: Changing the Viewpoint, Moving the Cube

After all pairs have completed Part I, ask for volunteers to comment on how these sketches compared with one another. Students should see that the projection changes if either the observer or the cube is at a different position. For example, when the observer moves, she or he may see different faces of the cube. In particular, students should see that the drawing is larger when the cube is closer to the screen. This should sharpen students' intuition about how the picture changes as distances change.

Ask, **How is the fact that the picture changes when the viewpoint changes related to the unit problem?** Students should see that because of this fact, they will need to decide on a specific location for the observer, in relation to the screen and the object, when they write a program for the unit problem. That location is called the *viewpoint* or the *center of the projection*.

Part II: Rotating the Cube

When most pairs have completed Part II, ask for volunteers to comment on their results.

Students should see that the drawing of the rotated cube is not simply a rotation of the original drawing. That is, they cannot simply turn partner A's drawing from Question 5 and make it match up with the drawing from Question 6.

This point is somewhat subtle but very important for the development of the final program for the unit. Take care to ensure that students appreciate the distinction.

For example, in addition to having students see that these two drawings are not simply rotations of each other, hold up a particular drawing of a cube (on plastic or paper) and then turn the drawing. Ask, **Is this what a cube rotating in three dimensions should look like?** You may want to then have students go back to their actual drawings to confirm that simply rotating a drawing does not give the right result.

As with Part I, ask the class how this fact—that the picture does more than simply rotate as the cube rotates—affects the unit problem. **What needs to rotate in the unit problem—the cube or the picture on the screen?** Help students to see that this means that when they write their programs, they must create a new projection for each small rotation of the cube. In other words, they will set up their cube in its initial position, draw the projection, rotate the cube and find the coordinates of the vertices of the rotated cube, draw the projection of that new cube, and so on. *Important:* Students must realize that it is not correct simply to make one projection and then turn that two-dimensional picture.

Projection Using Coordinates: An Overview

Ask students to think about the projection process they have done visually with the plastic and imagine doing this abstractly in terms of a three-dimensional coordinate system. That is, they should contemplate this scenario:

- Instead of having a physical cube, they have been given the *coordinates* for each corner of the cube.
- Instead of placing their head in some position and looking at the cube, they have been given *coordinates* for the viewer's eye.
- Instead of having a sheet of plastic to represent the screen, they have been given *an equation* for the screen.

Ask, **Given only this information, how would you determine where on the screen to draw each corner of the cube?** Students will actually be doing the arithmetic and algebra of this process in *Find Those Corners!*

For now, simply have them contemplate the idea briefly. You may want to start them thinking about their reflection for *Spiders and Cubes* by asking, **How is this process related to the *Lunch in the Window* problem?**

Key Questions

How is the fact that the picture changes when the viewpoint changes related to the unit problem?
What needs to rotate in the unit problem—the cube or the picture on the screen?

Given only this information, how would you determine where on the screen to draw each corner of the cube?
How is this process related to the *Lunch in the Window* problem?

Supplemental Activity

***Perspective on Geometry* (extension)** asks students to research the history of perspective in art.

Spiders and Cubes

Intent

Students reflect on how their recent work relates to the main unit problem.

Mathematics

The intent of this activity is for students to recognize that their work with lunches for spiders is a model for understanding *Cube on a Screen*.

Progression

Students explain what the recent problems involving lunches and spiders have to do with the unit problem. The subsequent discussion reviews the connections between *Lunch in the Window*, *Cube on a Screen*, and the unit problem.

Approximate Time

15 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

Individuals, followed by whole-class discussion

Doing the Activity

Students work on this activity independently.

Discussing and Debriefing the Activity

Let students share ideas with the class about the connection between the spider problems and the unit task of programming a turning cube on a screen.

Review *Cube on a Screen* and the discussion of how it relates to the situation in *Lunch in the Window*. To summarize:

- The viewpoint and a corner of the cube are like the two spiders.
- The line from the viewpoint to that corner is like the thread connecting the spiders.
- The screen is like the window.
- Figuring out where to place the corresponding dot on the screen is like finding the spot where the thread goes through the window.

Bring out as well that just as *Lunch in the Window* can serve as a model for understanding *Cube on a Screen, Cube on a Screen* is itself a model for the process of drawing a cube on the screen. In other words, students can think about the problem on three levels:

- The spider model
- The drawing on the clear plastic screen
- The drawing on the calculator screen

Find Those Corners!

Intent

Students express projections in terms of coordinates.

Mathematics

This activity requires students to make connections between the process of drawing the projection of a cube using their line of sight, as they did in *Cube on a Screen*, and the mathematics behind creating a projection without physically observing it, which they developed in another context in *Lunch in the Window*.

Progression

The teacher gives groups different viewpoints, and each group calculates and draws the projection of a cube onto a large sheet of graph paper based on its viewpoint. The groups post their drawings, and students try to determine which viewpoint matches which projection and describe their reasoning. In the subsequent discussion, the class develops and posts a procedure for finding the coordinates of a projected point.

Approximate Time

110 to 130 minutes

Classroom Organization

Small groups, followed by whole-class discussion

Materials

Grid chart paper (at least 1 sheet per group)
Cube (1 per group)
The three-dimensional coordinate systems groups made in *A Return to the Third Dimension*

Doing the Activity

Tell students that they are finally ready to do the numeric analysis involved in carrying out item 4: "Create a two-dimensional drawing of a three-dimensional object." Once they know how to make a two-dimensional projection of a stationary cube, they will focus on rotating the cube and then finishing the unit problem.

As groups begin work, assign a specific viewpoint to each group (see the list in the subsection "Suggested Viewpoints and Their Projections"). Tell groups not to list the viewpoints on their diagrams, because everyone will be trying to match viewpoints to the diagrams, based on specific details of the diagrams. If any groups finish early, assign them another viewpoint.

To save on grid chart paper, you might ask that students first do their work on their own paper. When they are satisfied with their work, they can transfer it to the chart paper.

Suggestions for Students

Because groups will be comparing their results, it will be helpful for everyone to use the same scale. Based on the suggested viewpoints listed below, 1-inch grid paper works well, with 2 inches per unit on both axes and both axes going from –2 to 4.

Suggest that students do the front face first and then work with the back face. They can label each two-dimensional projection with the three-dimensional coordinates of the corner of the cube it represents. Once they have drawn a projection point, they should connect it to all the other corner points on a common edge, using dotted lines for edges hidden behind visible faces.

You might also suggest that at least two students in each group independently find the coordinates of each projection and compare results before the group plots the point.

Suggested Viewpoints and Their Projections

Here are some viewpoints that work well with the scales just described:

- (4, 5, 10)
- (1, 1, 10)
- (1, 5, 10)
- (4, –3, 10)
- (–3, 5, 10)
- (–3, –4, 10)
- (1, –3, 10)
- (3, 1, 10)
- (4, 5, 20)
- (1, 1, 6)

Using 10 as the z-coordinate of the viewpoint makes the arithmetic come out nicely.

Post the list of viewpoints that you assign. When the activity is completed, students will post their diagrams showing the projected cubes (but not the viewpoints) and try to match each diagram with its associated viewpoint.

For Teachers: An Outline of the Process

For your reference, below is a sequence of steps for finding the projection of the vertex at (2, 2, 2) from the viewpoint (4, 5, 10), with the screen at $z = 5$, using the same reasoning as in *Lunch in the Window*. (The same screen location is used in the activity for all viewpoints.)

Note: Here, we are thinking of the projection as being partway from the vertex to the viewpoint. We could just as well think of it as being partway from the viewpoint to the vertex. In that case, the value of r would be $\frac{5}{8}$ instead of $\frac{3}{8}$, and the roles of the two points would be interchanged.

Students' first step will probably be to note that the z-distance from the vertex at (2, 2, 2) to the screen is 3 units and that the z-distance from (2, 2, 2) to the viewpoint (4, 5, 10) is 8 units. They should conclude that the projection of (2, 2, 2) is $\frac{3}{8}$ of the way from (2, 2, 2) to (4, 5, 10). In other words, they will find that for this case, the number represented as r in the spider and orchard problems comes out to $\frac{3}{8}$. Therefore, the x- and y-distances from (2, 2, 2) to its projection must be $\frac{3}{8}$ of the corresponding distances from (2, 2, 2) to the viewpoint.

The x-distance from (2, 2, 2) to (4, 5, 10) is 2, and $\frac{3}{8}$ of this is 0.75. So the x-coordinate of the projection is 2.75 (adding 0.75 to the x-coordinate of the vertex). Similarly, the y-distance from (2, 2, 2) to (4, 5, 10) is 3, and $\frac{3}{8}$ of this is 1.125. So the y-coordinate of the projection is 3.125. Therefore, the projection of (2, 2, 2) is the point (2.75, 3.125, 5).

Help students to articulate that the ratio $\frac{3}{8}$ in this example corresponds to the r-value in the formula developed in *And Fred Brings the Lunch*. More generally, this could be expressed as the ratio

$$\frac{z_{\text{screen}} - z_{\text{vertex}}}{z_{\text{viewpoint}} - z_{\text{vertex}}}$$

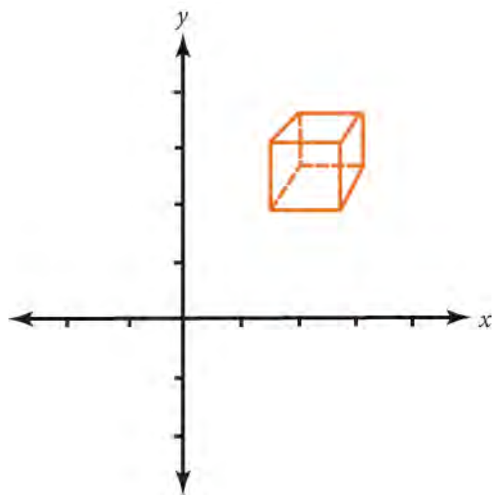
In drawing the cube on graph paper, students should realize that they need only look at the x- and y-coordinates of the projection. That is, the graph paper is assumed to represent the plane $z = 5$, so points plotted there will automatically have a z-coordinate of 5. Thus, for example, the projection of (2, 2, 2) should be plotted as the point (2.75, 3.125).

A Chart of Projected Coordinates

For each of the viewpoints listed previously, the chart here gives the x- and y-coordinates of the projected vertices and a diagram of the projected cube, showing the visible edges with solid lines and the hidden edges with dotted lines. The axes shown are the x- and y-axes within the screen, that is, on the plane $z = 5$. The chart shows what happens to each vertex. For instance, the first example shows that the point (0, 0, 0) is projected to the point (2, 2.5).

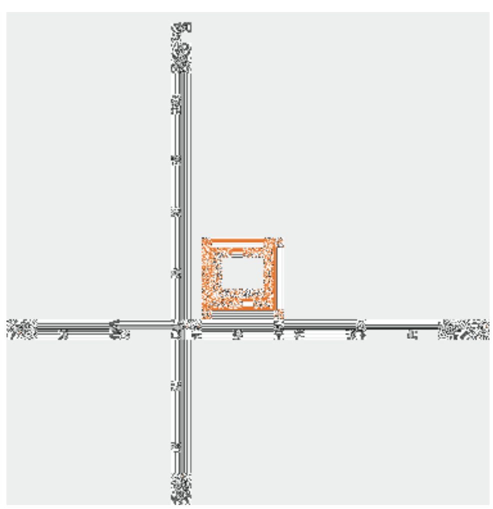
viewpoint: (4, 5, 10)

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 2 \\ 2 & 0 & 2 \\ 0 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 2.5 \\ 3 & 2.5 \\ 2 & 3.5 \\ 3 & 3.5 \\ 1.5 & 1.875 \\ 2.75 & 1.875 \\ 1.5 & 3.125 \\ 2.75 & 3.125 \end{bmatrix}$$



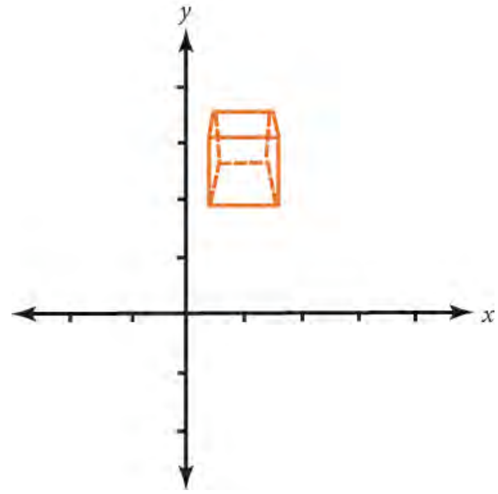
viewpoint: (1, 1, 10)

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 2 \\ 2 & 0 & 2 \\ 0 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} 0.5 & 0.5 \\ 1.5 & 0.5 \\ 0.5 & 1.5 \\ 1.5 & 1.5 \\ 0.375 & 0.375 \\ 1.625 & 0.375 \\ 0.375 & 1.625 \\ 1.625 & 1.625 \end{bmatrix}$$



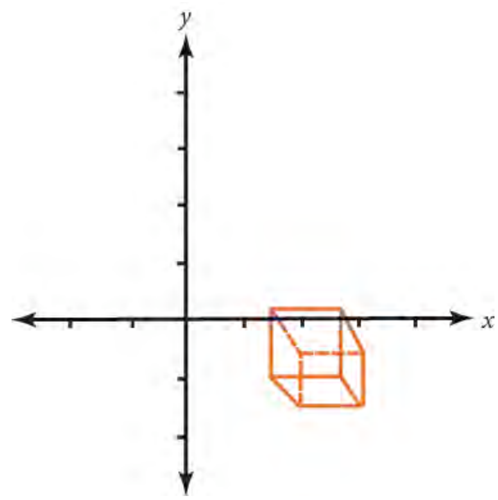
viewpoint: (1, 5, 10)

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 2 \\ 2 & 0 & 2 \\ 0 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} \to \begin{bmatrix} 0.5 & 2.5 \\ 1.5 & 2.5 \\ 0.5 & 3.5 \\ 1.5 & 3.5 \\ 0.375 & 1.875 \\ 1.625 & 1.875 \\ 0.375 & 3.125 \\ 1.625 & 3.125 \end{bmatrix}$$



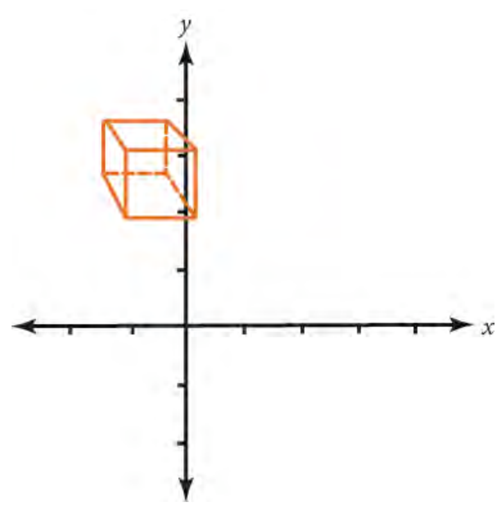
viewpoint: (4, –3, 10)

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 2 \\ 2 & 0 & 2 \\ 0 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} \to \begin{bmatrix} 2 & -1.5 \\ 3 & -1.5 \\ 2 & -0.5 \\ 3 & -0.5 \\ 1.5 & -1.125 \\ 2.75 & -1.125 \\ 1.5 & 0.125 \\ 2.75 & 0.125 \end{bmatrix}$$



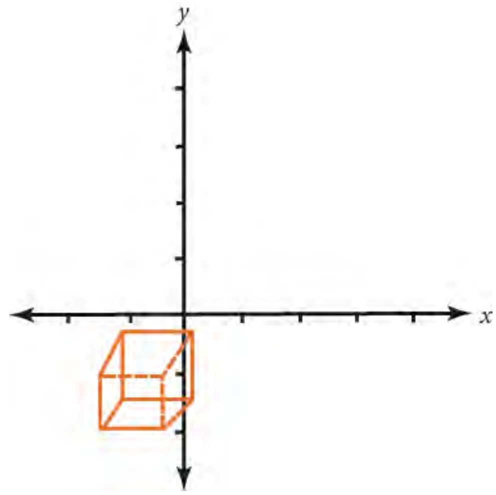
viewpoint: (–3, 5, 10)

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 2 \\ 2 & 0 & 2 \\ 0 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} \to \begin{bmatrix} -1.5 & 2.5 \\ -0.5 & 2.5 \\ -1.5 & 3.5 \\ -0.5 & 3.5 \\ -1.125 & 1.875 \\ 0.125 & 1.875 \\ -1.125 & 3.125 \\ 0.125 & 3.125 \end{bmatrix}$$



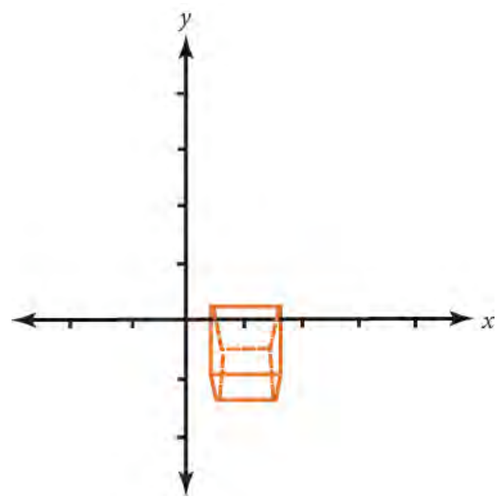
viewpoint: (−3, −4, 10)

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 2 \\ 2 & 0 & 2 \\ 0 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} -1.5 & -2 \\ -0.5 & -2 \\ -1.5 & -1 \\ -0.5 & -1 \\ -1.125 & -1.5 \\ 0.125 & -1.5 \\ -1.125 & -0.25 \\ 0.125 & -0.25 \end{bmatrix}$$



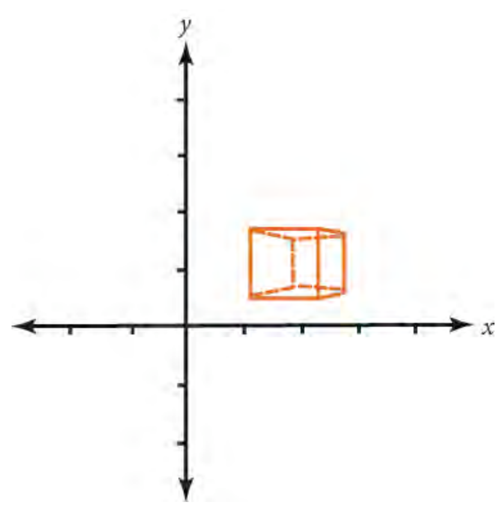
viewpoint: (1, −3, 10)

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 2 \\ 2 & 0 & 2 \\ 0 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} 0.5 & -1.5 \\ 1.5 & -1.5 \\ 0.5 & -0.5 \\ 1.5 & -0.5 \\ 0.375 & -1.125 \\ 1.625 & -1.125 \\ 0.375 & 0.125 \\ 1.625 & 0.125 \end{bmatrix}$$



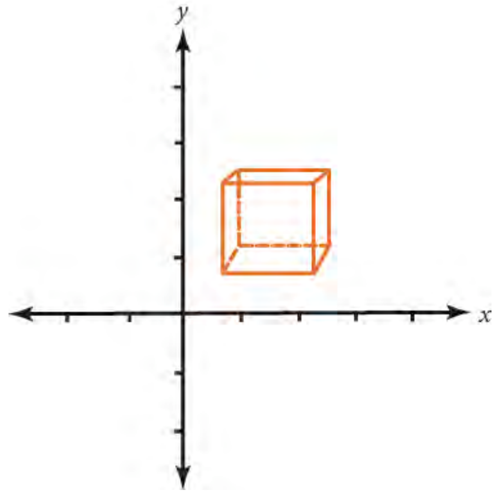
viewpoint: (3, 1, 10)

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 2 \\ 2 & 0 & 2 \\ 0 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} 1.5 & 0.5 \\ 2.5 & 0.5 \\ 1.5 & 1.5 \\ 2.5 & 1.5 \\ 1.125 & 0.375 \\ 2.375 & 0.375 \\ 1.125 & 1.625 \\ 2.375 & 1.625 \end{bmatrix}$$



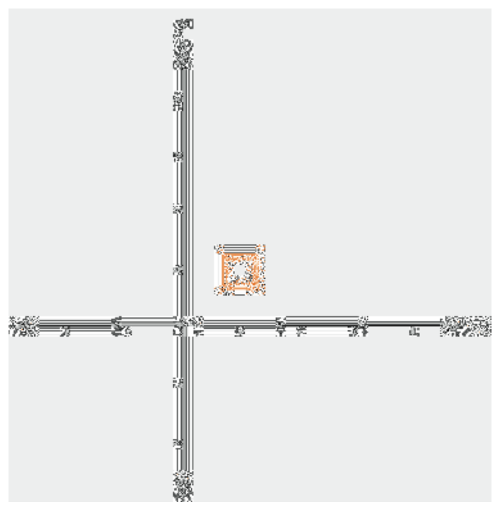
viewpoint: (4, 5, 20)

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 2 \\ 2 & 0 & 2 \\ 0 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1.25 \\ 2.5 & 1.25 \\ 1 & 2.75 \\ 2.5 & 2.75 \\ 0.667 & 0.833 \\ 2.333 & 0.833 \\ 0.667 & 2.5 \\ 2.333 & 2.5 \end{bmatrix}$$



viewpoint: (1, 1, 6)

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 2 \\ 2 & 0 & 2 \\ 0 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} 0.833 & 0.833 \\ 1.167 & 0.833 \\ 0.833 & 1.167 \\ 1.167 & 1.167 \\ 0.75 & 0.75 \\ 1.25 & 0.75 \\ 0.75 & 1.25 \\ 1.25 & 1.25 \end{bmatrix}$$



Discussing and Debriefing the Activity

Use your observations as a guide to whether a presentation on the numeric details is needed. If possible, focus the discussion on the process of matching diagrams and viewpoints.

If groups haven't already done so, have them display their diagrams. Remind them not to list the viewpoints they used. Give everyone 5 to 10 minutes to try to match the diagrams with the posted list of viewpoints.

Focus the discussion on how students made the matches. (We are assuming they didn't actually go through the projection for every viewpoint.) Have volunteers point to a specific poster (other than their own) and explain how they figured out which viewpoint it represents.

For example, the first diagram in the chart above shows that the viewpoint is above and to the right of the cube, because the front, right, and top faces of the cube are visible. The only viewpoints in the list that fit this condition are (4, 5, 10) and (4, 5, 20). Of these two, the first gives a perspective more from the side, because it is closer, while (4, 5, 20) shows more of a head-on view of the cube, because it is farther away.

Similarly, from the viewpoint (1, 5, 10), one should see only the front and top faces, and there is only one diagram that fits this condition.

You may not need to go over the computational details for any of the examples, depending on how comfortable students seem with the computations. Following the matching activity, students will develop a general outline for how to do this, and you can review the computational details as part of that work.

One detail you may wish to raise now is that although the cube has eight vertices, there are only two r-values in the computations. That is, the vertices on the front face all give one value for the ratio $\frac{\text{distance to screen}}{\text{distance to viewpoint}}$, while the vertices on the back face all give a second value for this ratio.

Ask students if this is always the case for every choice of screen, viewpoint, and cube position. **Are there always only two *r* values?** They should see that this holds true only when the cube is positioned with two faces parallel to the screen.

How to Project

Now ask students to work in their groups to write out algebraic steps for projecting a point onto the screen. For uniformity of notation, you may want to suggest that they represent the point being projected as (x_1, y_1, z_1), the viewpoint as (x_2, y_2, z_2), and the screen as the plane $z = k$.

When students seem ready, bring them together to develop a class outline for this process. Tell them they will use this outline for writing the final program (in *The Turning Cube Outline*).

Here is a sample outline. Again, we are thinking of the screen as being partway from the vertex to the viewpoint, rather than partway from the viewpoint to the vertex.

Step 1. Find the ratio *r*. Find the difference in z-coordinates between the projected point and the screen, find the difference in z-coordinates between the projected point and the viewpoint, and find the ratio of these two z-distances. This gives

$$r = \frac{k - z_1}{z_2 - z_1}$$

Step 2. Find the *x*-coordinate of the projection. Find the difference in *x*-coordinates between the projected point and the viewpoint, multiply this *x*-distance by *r*, and add the result to the *x*-coordinate of the point. This gives

$$x_{\text{projection}} = x_1 + r(x_2 - x_1)$$

Step 3. Find the *y*-coordinate of the projection. Find the difference in *y*-coordinates between the projected point and the viewpoint, multiply this *y*-distance by *r*, and add the result to the *y*-coordinate of the point. This gives

$$y_{\text{projection}} = y_1 + r(y_2 - y_1)$$

Step 4. Plot the point. Plot the point using the *x*- and *y*-coordinates found in steps 2 and 3.

Post this description prominently. You can now check off item 4 from the unit plan from the discussion introducing *Picture This!*: "Create a two-dimensional drawing of a three-dimensional object."

In the next section, *Rotation in Three Dimensions*, students will look at item 5: "Change the position of an object located in a three-dimensional coordinate system." As noted in the discussion introducing *Goin' Round the Origin*, this will involve looking only at rotations. In *The Turning Cube Outline*, students will look at the task of putting all the pieces together.

Key Question

Are there always only two *r* values?

Supplemental Activity

Let the Calculator Do It! (extension and reinforcement) challenges students to write a program to find the projection of a point onto the plane $z = 5$, given the viewpoint.

An Animated Outline

Intent

Students prepare outlines for their animation projects.

Mathematics

Students now complete the outline of their program for *POW 9: An Animated POW.*

Progression

Students consult with their partners to complete the outlines for their projects.

Approximate Time

30 to 45 minutes for activity (at home, with partners)

Classroom Organization

Partners

Doing the Activity

Students will need to consult with their partners on this activity.

Discussing and Debriefing the Activity

Collect students' outlines and check them for problems. You may want to remind students to keep written copies of their programs and to leave enough time for debugging.

Mirrors in Space

Intent

Students express reflections in three dimensions using matrices.

Mathematics

This activity is a three-dimensional analog of *Flipping Points.*

Progression

Students find the reflection of a given point through the *yz*-plane, generalize their work, and find a matrix that will do the same work.

Approximate Time

25 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

Individuals, then groups, followed by whole-class discussion

Doing the Activity

This activity requires no introduction.

Discussing and Debriefing the Activity

Have students compare answers in their groups, and then ask students from one or more groups report on each question.

The use of the specific point (4, 6, 2) in Question 1a will help students see what is happening with the coordinates. In general, the point (x, y, z) is reflected to $(-x, y, z)$.

Question 2 here is similar to Question 3 of *Flipping Points*. If needed, suggest that students write out the question in matrix form, something like the equation below.

$$\begin{bmatrix} x & y & z \end{bmatrix} \cdot \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix} = \begin{bmatrix} -x & y & z \end{bmatrix}$$

This should lead them to see that the reflection matrix is

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

Where Are We Now?

Intent

Students review where they are in the unit.

Mathematics

Students review the main ideas of the unit as they reflect upon the key mathematical concepts and how they relate to the unit problem.

Progression

In the discussion, students share their perceptions from the beginning of the unit, their views about the key mathematical ideas, and their thoughts about what's still needed.

Approximate Time

25 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

Individuals, followed by whole-class discussion

Doing the Activity

Students work on this activity individually.

Discussing and Debriefing the Activity

Ask various students to contribute what they thought the unit would entail and what mathematics they thought they would be learning. Then ask other students to explain the important ideas of the unit so far and how they relate to the unit problem.

Finally, to get an idea of where the class stands in the process of solving the unit problem, ask for volunteers to contribute thoughts on what they have to do next.

This activity is a good one to collect to get a sense of how well students understand the unit so far.

Rotation in Three Dimensions

Intent

In these activities, students complete the solution of the unit problem.

Mathematics

These activities help students to adapt what they know about rotation in two dimensions to writing matrices for rotating in three dimensions. Students then develop an outline for the turning cube program and write or analyze programming code to accomplish the task.

Progression

In *Follow That Point!* and *One Turn of a Cube,* students apply what they have learned about rotation to find the coordinates of points rotated about the z-axis. They write matrices for accomplishing rotations about each of the three axes in *Rotation Matrix in Three Dimensions*.

With all of the various pieces finally in place, students put it all together to outline a program for turning the unit problem's cube in *The Turning Cube Outline*.

Follow That Point!
One Turn of a Cube
Rotation Matrix in Three Dimensions
The Turning Cube Outline
Beginning Portfolio Selection

Follow That Point!

Intent

Students find the coordinates of a point rotated around the *z*-axis.

Mathematics

This activity helps students to visualize rotations about an axis in three dimensions. Students express rotation in three dimensions of both a point and a line segment in terms of coordinates.

Progression

The teacher introduces the activity by having students review rotations in two dimensions and by using a physical model to help them picture what it means to rotate a cube around the *z*-axis. Students then consider where a specified point will end up after a 20° rotation around the *z*-axis and where a given line segment will end up after the same rotation. The subsequent discussion emphasizes that rotation around the *z*-axis is like rotation in the *xy*-plane, with the *z*-coordinate kept fixed.

Approximate Time

40 minutes

Classroom Organization

Small groups or individuals, preceded and followed by whole-class discussion

Materials

Large foam cube (or similar object) and two sticks for a classroom demonstration
Cubes (1 per student)

Doing the Activity

Tell students that they will now look at the last piece of the grand plan, item 5, as modified in the discussion introducing *Goin' Round the Origin:*

5. **Change the position of an object located in a three-dimensional coordinate system.**
    - **Rotate the object.**

Review Rotation in Two Dimensions

Ask for one or more volunteers to review what the class knows about rotation in two dimensions. Elicit statements in terms of both coordinates and matrices.

In terms of coordinates, you will already have something like this posted from the discussion of *Comin' Round Again* (*and Again...*):

> **If a point (x, y) is rotated counterclockwise through an angle ϕ, then the coordinates (x', y') of its new location are given by the equations**
>
> $$\mathbf{x' = x \cos \phi - y \sin \phi}$$
> $$\mathbf{y' = x \sin \phi + y \cos \phi}$$

In terms of matrices, you will already have something like this posted from the discussion of *More Memories of Matrices:*

> **To rotate a matrix of two-dimensional points counterclockwise around the origin by an angle ϕ, multiply that matrix on the right by the matrix**
>
> $$\begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}$$

What Is a Rotation in Three Dimensions?

The next task is to be sure students have a visual picture of what is meant by rotation in three dimensions about an axis. As described and diagrammed in the discussion preceding *Picture This!*, we are imagining the board as the *xy*-plane, with the positive *z*-axis coming "forward." Place a cube "in space" and ask students for the coordinates of its vertices. It may help to use a stick attached to a foam cube to demonstrate the process, as described in that earlier discussion.

Ask, **What will happen to the coordinates if the object is rotated around the *z*-axis?** Each student should have a small cube to work with to get a sense of what is happening.

Let various students come up to show the rotation. They should see that the *z*-coordinate of each point stays the same, because each point is traveling within a plane parallel to the *xy*-plane. Thus, students can essentially ignore the *z*-coordinate and treat the motion of each point within that plane.

Follow That Point!

Let groups begin work. If they struggle with Question 1, you might suggest they imagine the graph paper as if it were in the plane $z = -5$. They can then think of a point as rotating around the origin on the paper.

Discussing and Debriefing the Activity

Ask several students to present their answers and reasoning.

Students should see that each of the two points under consideration is being rotated in a plane parallel to the *xy*-plane. That is, the point (2, 3, −5) is rotating within the plane $z = -5$, and the point (5, 2, 6) is rotating within the plane $z = 6$. Thus, students can work with the *x*- and *y*-coordinates as if the points were in the *xy*-plane itself.

If (2, 3) were rotated counterclockwise 20° around the origin, it would move to (2 cos 20° − 3 sin 20°, 2 sin 20° + 3 cos 20°). Therefore, when (2, 3, −5) is rotated counterclockwise 20° around (0, 0, −5), within the plane $z = -5$, it moves to (2 cos 20° − 3 sin 20°, 2 sin 20° + 3 cos 20°, −5), or approximately (0.85, 3.50, −5).

Note: If students go back to basics, they might express their answer to Question 1 as something like this:

$$\left(\sqrt{13}\cos\left[\tan^{-1}\left(\frac{3}{2}\right)+20^\circ\right],\sqrt{13}\sin\left[\tan^{-1}\left(\frac{3}{2}\right)+20^\circ\right],-5\right)$$

If anyone does this, use the occasion to review why this approach doesn't necessarily work in all quadrants. This method would give the same result if students started with (−2, −3, −5) instead of (2, 3, −5). You might also review how the rotation formula was developed, using the sine-of-a-sum and cosine-of-a-sum formulas.

Question 2

The main goal of Question 2 is to bring out that the result of this rotation can be described as the segment whose endpoints are the results of rotating the original endpoints.

Students should be able to find the result of rotating (5, 2, 6) just as they did with (2, 3, −5). A 20° counterclockwise rotation about the *z*-axis takes the point (5, 2, 6) to the point (5 cos 20° − 2 sin 20°, 5 sin 20° + 2 cos 20°, 6), or approximately (4.01, 3.59, 6). Thus, the segment connecting (2, 3, −5) and (5, 2, 6) ends up after rotation as the segment connecting (0.85, 3.50, −5) and (4.01, 3.59, 6).

Key Question

What will happen to the coordinates if the object is rotated around the *z*-axis?

One Turn of a Cube

Intent

Students find the coordinates of the vertices of a cube after rotation around the z-axis.

Mathematics

This activity continues the work with rotations in three dimensions.

Progression

Students identify the coordinates of the vertices of a cube that is placed in 3-space, find the coordinates of one of the faces after a given rotation of the cube around the z-axis, and sketch the cube in each of the two positions. The subsequent discussion emphasizes that the z-coordinates stay fixed. Students should save their results to use during the discussion of *Rotation Matrix in Three Dimensions*.

Approximate Time

30 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

Individuals, then groups, followed by whole-class discussion

Doing the Activity

This activity needs no introduction.

Discussing and Debriefing the Activity

Have students confer in their groups. It is important for them to see that the z-coordinate stays fixed and that the rotation is like a two-dimensional rotation involving only the x- and y-coordinates. Each point moves around in a two-dimensional plane of the form $z = k$.

To illustrate the algebra of the rotation, have students suppose one of the vertices of the cube is at (3, 5, 7). The rotated point will still have a z-coordinate of 7. The new x- and y-coordinates, labeled here as x' and y', are found by the usual formulas:

$$x' = x \cos 30° - y \sin 30°$$
$$y' = x \sin 30° + y \cos 30°$$

In other words, the rotated point has these new coordinates:

$$(3 \cos 30° - 5 \sin 30°, 3 \sin 30° + 5 \cos 30°, 7)$$

or approximately (0.10, 5.83, 7). Have students verify that this seems reasonable on the graph.

If students placed their cubes so that some vertices were on the z-axis, they should have seen that these vertices did not move when the cube was rotated around the z-axis.

Have students save their results to use during the discussion of *Rotation Matrix in Three Dimensions*.

Rotation Matrix in Three Dimensions

Intent

Students use matrices to express rotations in 3-space.

Mathematics

With the understanding of rotations in three dimensions gained in *Follow That Point!*, students are now ready to express those rotations using matrices.

Progression

Students find the matrices for rotation around each of the three axes. During the follow-up discussion, they use their matrix for rotation around the z-axis to check their results from *One Turn of a Cube*.

Approximate Time

35 minutes

Classroom Organization

Small groups or individuals, followed by whole-class discussion

Doing the Activity

This activity requires no introduction.

Discussing and Debriefing the Activity

Have a student report his or her results. Students should be able to figure out that the rotation matrix [B] should be

$$[B] = \begin{bmatrix} \cos 30^\circ & \sin 30^\circ & 0 \\ -\sin 30^\circ & \cos 30^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ask, **Can you verify that this works not just for a single point, but also for a matrix of three-dimensional points?** That is, students should take their own vertex matrices from *One Turn of a Cube,* multiply them by this matrix B, and see that they do get the coordinates of the rotated vertices. Even if they did not develop the matrix B on their own, they should see that it does the job.

Before looking at rotations around other axes, ask students to generalize Question 1 for an arbitrary angle of rotation. Post a general statement something like this:

> **To rotate a matrix of three-dimensional points counterclockwise around the *z*-axis by an angle ϕ, multiply that matrix on the right by the matrix**
>
> $$\begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

For the issue of rotation around other axes, students may see that they simply need to rearrange the entries of the rotation matrix. For example, for rotation around the *x*-axis, the matrix looks like this:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$

For rotation around the *y*-axis, one multiplies the matrix on the right by

$$\begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix}$$

Key Question

Can you verify that this works not just for a single point, but also for a matrix of three-dimensional points?

The Turning Cube Outline

Intent

Students solve the unit problem.

Mathematics

After developing an outline for the program for the turning cube, the class creates or analyzes a plain-language program and programming code.

Progression

Students develop an outline for a program to turn the cube. Subsequently, the class concludes the unit using one of several teacher options.

Approximate Time

5 minutes for introduction
30 minutes for activity (at home or in class)
50 minutes for discussion and concluding activity (time may vary widely, depending upon the option that is selected for concluding the unit)

Classroom Organization

Individuals, then groups, preceded by whole-class discussion and followed by small-group or whole-class activity

Doing the Activity

This activity is students' final preparation for solving the unit problem. You may want to review the generic animation outline developed during the discussion of *Move That Line!*:

Setup program
Set the initial coordinates
Start the loop
- Clear the screen
- Draw the next figure
- Delay
- Change the coordinates

End the loop

You may want to remind students that along with setting the initial conditions, they also need to set the "change matrix," which would be the matrix of rotation.

Discussing and Debriefing the Activity

Let groups work together, sharing ideas for a few minutes, and then bring the class together to develop a common outline. As a first stage, you might get something like this:

> Setup program
> Set the coordinates for the viewpoint and for the initial vertices of the cube, set the equation for the screen, and set the rotation matrix
> Start the loop
> - Clear the screen
> - Find the coordinates for the projections on the screen of the vertices of the cube
> - Draw the projected cube by connecting certain pairs of vertices
> - Delay
> - Change the coordinates of the vertices of the cube using a three-dimensional rotation matrix
>
> End the loop

The most subtle element of this outline is the fact that the vertices of the *actual* cube must be rotated, not merely the vertices of the *projected* cube. You may want to return to students' work on *Cube on a Screen* to help them clarify why this is necessary.

You may also want to remind students that in the discussion following *Find Those Corners!*, they developed fairly detailed instructions for the step "Find the coordinates for the projections on the screen of the vertices of the cube."

Concluding "As the Cube Turns"

There are at least three options for concluding the unit:

- Let students work in their groups or as a class to write a program to turn the cube.
- Give students the program TURNCUBE and have them explain how it works.
- Have students explain TURNCUBE and then work on developing an improved version of the program.

Suggestions for each option are given below. Your choice may depend, in part, on how well students have understood the mechanics of programming and on your judgment about whether having them develop the details of the program on their own will enhance their understanding or obscure it. You may want to give students a voice in this decision.

Although this concluding task of the unit can be completed in one class period, you may decide it is worthwhile to allow a second day for students to finish this work. IMP field-test teachers have used all three options. Even when they have used the second or third option, in which students do not actually write the program to turn the cube, their students have felt it was a satisfactory conclusion to the unit.

If students write the program: If you want students to work in groups to write their own program, have them write it on paper before putting it into the calculator. If a computer-to-calculator link is available (so programs can be written on a computer and transferred to the calculator), we strongly recommend that students use it. The computer makes entering and debugging programs easier, and it can store the programs overnight.

One suggestion for simplifying the programming is for students to find the two ratios for the front and back faces of the cube numerically and simply use these numbers in the program, rather than compute them as part of the program. This assumes students will place the cube so that two of its faces are parallel to the screen.

If students focus on explaining the program: If you want to focus on having students simply explain the program TURNCUBE, here are some specific tasks to give them:

- Find the initial position of the cube.
- Find the plane of the screen.
- Find the angle for each rotation.
- Explain the roles of specific variables.

If students develop an improved version of the program: You may want to discuss as a class how the program might be improved. For instance, it might provide for user input on the choice of viewpoint, screen, and position of the cube.

Key Question

What are the numeric ratios for the front and back faces?

Beginning Portfolio Selection

Intent

Students begin to complete the unit portfolio.

Mathematics

Students start compiling their portfolios for this unit by summarizing how they learned about three of the items on the outline for the unit problem.

Progression

Students choose and discuss an activity for three of the five items on the original outline for the unit problem. They will complete this work for the remaining two items in *Continued Portfolio Selection*.

Approximate Time

30 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

Individuals, followed by whole-class discussion

Doing the Activity

This activity requires no introduction.

Discussing and Debriefing the Activity

Have a couple of volunteers read their descriptions of how each item from the outline fit into the development of the unit. You may also want to discuss what makes a good description.

Also have some students state their choices of activities and explain how these activities helped them understand the ideas involved.

Because students will be working further with the unit outline from this activity (see *Continued Portfolio Selection*), have them identify clearly which activity they selected goes with which item in the outline. Although this may seem obvious to them now, it may not be as clear in a couple of days.

Creating Animations

Intent

In these activities, students complete the unit.

Mathematics

The main activity in these activities is completion of *POW 9: An Animated POW*. As they prepare for the unit assessments, students also complete their unit portfolio.

Progression

An Animated POW Write-up
Continued Portfolio Selection
As The Cube Turns Portfolio

An Animated POW Write-up

Intent

Students complete their write-ups for *POW 9: An Animated POW*.

Mathematics

This activity asks students to prepare a written copy of their animation programs and to develop their presentations.

Progression

Assign this activity after students have their final workday for their animation POW (see the subsection "Schedule for the POW" in the discussion notes for *POW 9: An Animated POW*) and before their presentations of their programs.

Approximate Time

30 to 45 minutes for activity (at home or in class)

Classroom Organization

Individuals

Doing the Activity

Partners will need to collaborate on their write-ups. They can either prepare individual write-ups or work together on a single write-up and make a photocopy of it.

Tell students that their write-ups for *POW 9: An Animated POW* will be part of their portfolios for this unit.

Discussing and Debriefing the Activity

No discussion of this activity is necessary.

Continued Portfolio Selection

Intent

Students continue their work on the unit portfolio.

Mathematics

Students continue summarizing how they learned about each item in the outline for the unit problem.

Progression

Beginning Portfolio Selection asked students to choose and discuss an activity for each of three of the five items in the original outline for the unit problem. Students now complete this work for the remaining two items.

Approximate Time

20 to 25 minutes for activity (at home or in class)
0 to 5 minutes for discussion

Classroom Organization

Individuals, followed by optional whole-class discussion

Doing the Activity

No introduction is needed for the activity.

Discussing and Debriefing the Activity

Because different students likely chose different items from the outline for *Beginning Portfolio Selection,* you may have covered all five items in the previous discussion and may not need to have any discussion of the current activity. As with *Beginning Portfolio Selection,* have students identify clearly which activity they selected goes with which item in the outline.

As The Cube Turns Portfolio

Intent

Students reflect upon the unit's key concepts as they complete their unit portfolios and write their cover letters.

Mathematics

In the portfolio cover letter, students describe the main mathematical ideas of the unit, how they were developed, and how they were used to solve the unit problem.

Progression

Students select activities for inclusion in their portfolios and write their cover letters. Following this activity, sharing of the cover letters is used to introduce a review of the unit. Viewing a video on animation is a nice way to complete the unit.

Approximate Time

30 to 40 minutes for activity (at home or in class)
10 minutes for discussion

Classroom Organization

Individuals, followed by whole-class discussion

Materials

Optional: A video on the making of animation features

Doing the Activity

Be sure students bring back their portfolios tomorrow with the cover letter as the first item. They should also bring to class any other work they think will be of help on the unit assessments.

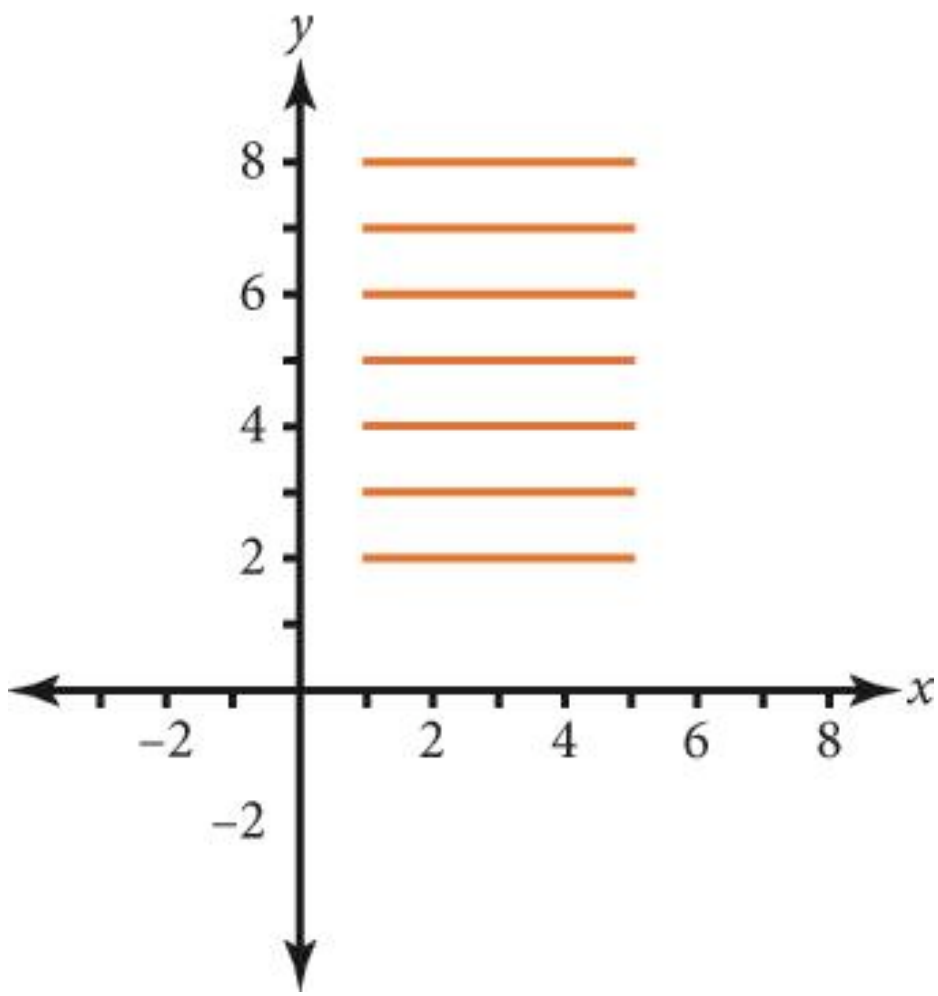
Discussing and Debriefing the Activity

Have students share their cover letters as a way to start a summary discussion of the unit. Then let them brainstorm ideas of what they have learned in this unit. This is a good opportunity to review terminology and to place this unit in a broader mathematics context.

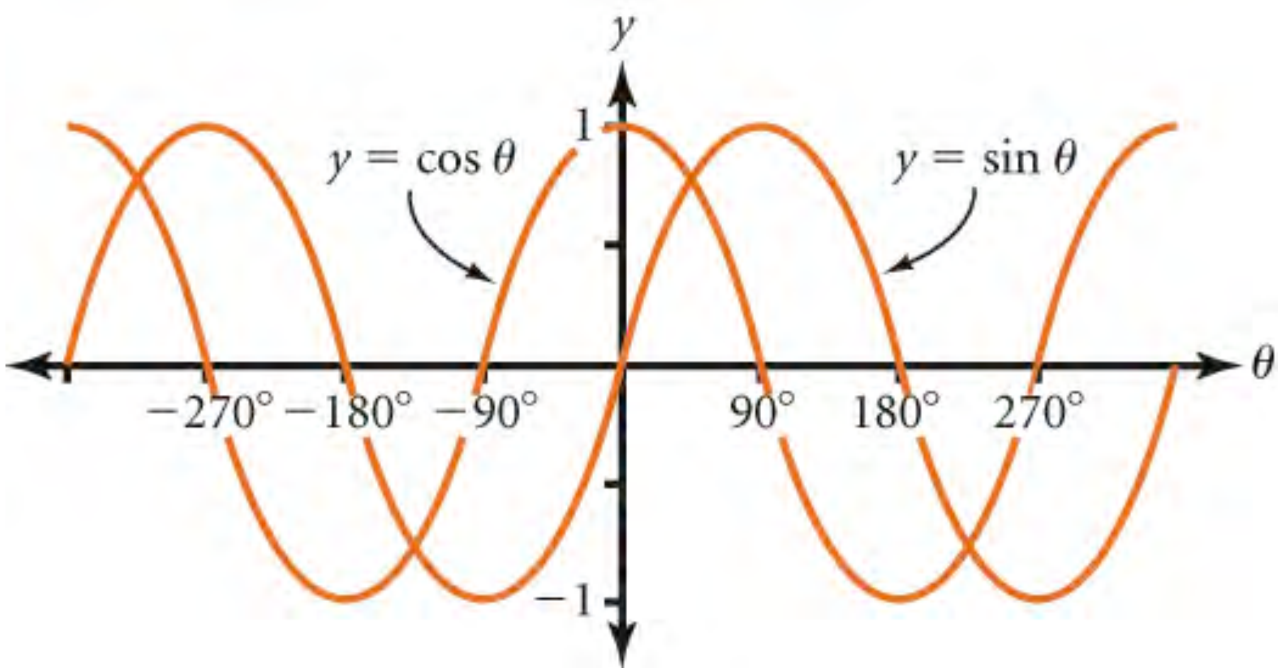
If you haven't yet shown a video on the making of a major animation feature film, you may want to now. A good video can help students see that professional animators really do use the mathematics they have learned.

Blackline Master

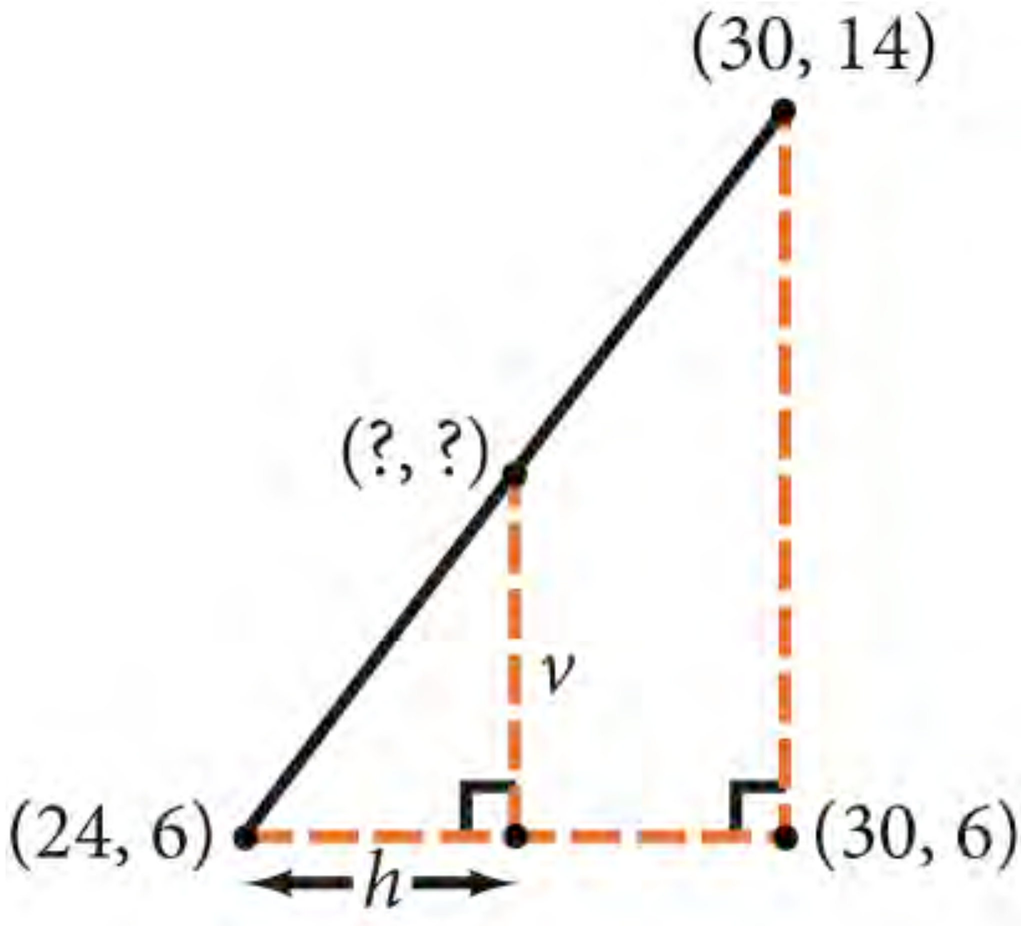
Learning the Loops

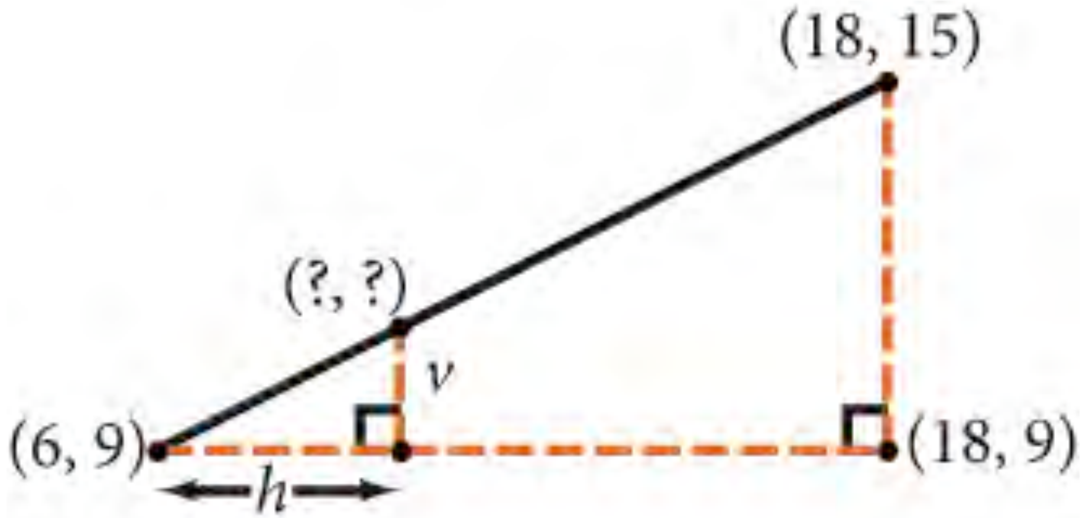


Oh, Say What You Can See

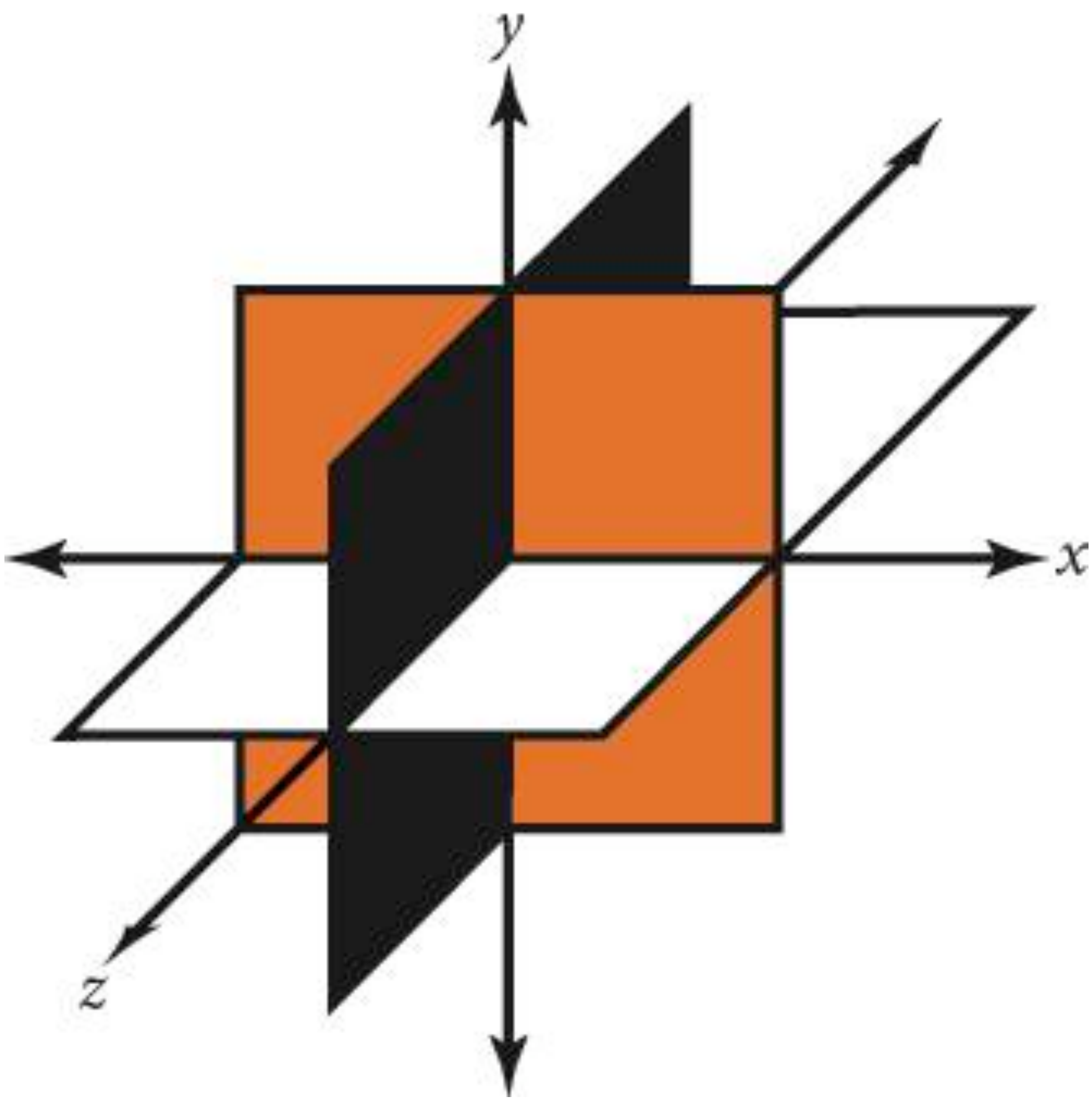


“A Snack in the Middle” Revisited




(18, 15)
(?, ?)
v
(6, 9)
(18, 9)
h

A Return to the Third Dimension



¼-Inch Graph Paper

1-Centimeter Graph Paper

1-Inch Graph Paper

Assessments

In-Class Assessment

Here is a plain-language program for a graphing calculator:

Program: DRAW

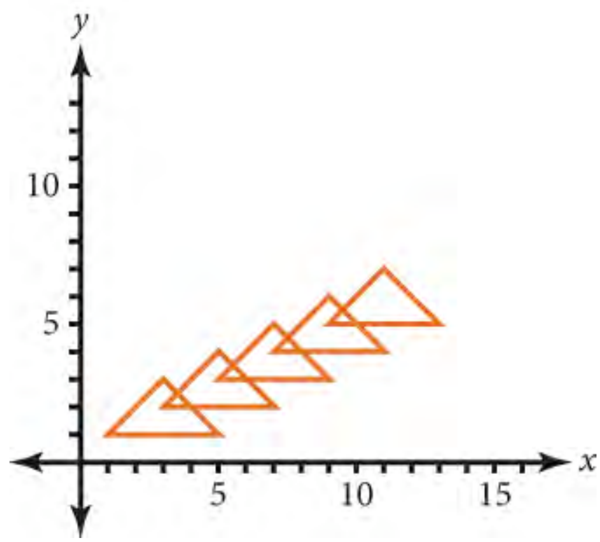
Setup program

Let A be the matrix $\begin{bmatrix} 1 & 1 \\ 3 & 3 \\ 5 & 1 \end{bmatrix}$

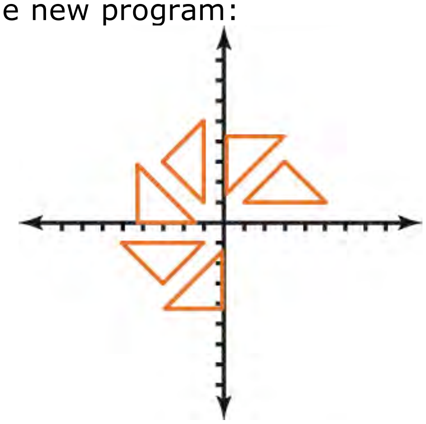
- Draw a line from (a_{11}, a_{12}) to (a_{21}, a_{22})
- Draw a line from (a_{11}, a_{12}) to (a_{31}, a_{32})
- Draw a line from (a_{31}, a_{32}) to (a_{21}, a_{22})

1. On graph paper, draw what the screen should look like after running this program using an appropriate viewing window. Show scales for the axes.

2. Use a loop and a matrix B to modify the program DRAW so the screen will look like this after running the new program:



3. Use a loop and a matrix C to modify the program DRAW so the screen will look like this after running the new program:



Note that the scales in this diagram are the same as in Question 2, although the viewing window is different. The numbers on the scales have been omitted to avoid cluttering the diagram.

Take-Home Assessment

Part I: Where Are We?

This pair of equations will project a point (u, v, w) onto a screen parallel to the xy-plane so that the x- and y-coordinates of the projected point are a and b.

$$a = u + \left(\frac{6-w}{2-w}\right)(0-u)$$

$$b = v + \left(\frac{6-w}{2-w}\right)(1-v)$$

1. What are the coordinates of the viewpoint for the projection?

2. What is the equation of the screen?

3. Explain in detail why the equations for a and b give the projection you describe.

Part II: How Does It Turn?

You have seen that when a point (x, y) is rotated counterclockwise around the origin through an angle ϕ, the new point has coordinates $x \cos \phi - y \sin \phi$ and $x \sin \phi + y \cos \phi$.

Explain how those two formulas were developed.

As the Cube Turns Calculator Guide for the TI-83/84 Family of Calculators

The unit *As the Cube Turns* is by far the most calculator-intensive unit of the IMP curriculum. In this unit, students learn to program an animation of a three-dimensional object. In doing so, they become proficient with several basic programming commands and techniques. However, the unit is much more than simply a programming tutorial. The unit problem requires the use of matrix manipulation, coordinate geometry, geometric transformations and projections, and trigonometric identities, as well as the ability to use a technical manual.

If you do not have previous experience with programming, this unit may seem intimidating at first. To help you, sample programs are included here for nearly every programming assignment. Ultimately, however, you will become most comfortable simply by allowing yourself to become a student again, making up your own programs for the assignments and learning through your own mistakes and accomplishments. Don't be afraid to let your students know that you are learning, too. It will add significantly to their own sense of accomplishment.

Before you know it, you will be just as eager as your students to display your latest creation to the class. After all, that's what IMP is about—the thrill of discovery!

Be sure to work out some system through which your students can take calculators home overnight. Students who have access to calculators outside of class will have a tremendous advantage over those who do not. In fact, they will usually get caught up in their programming projects and spend far more time researching and practicing their programming techniques than you would ever dream of requiring of them.

Picture This!: You will need to have the program TURNCUBE, or something similar, entered into your overhead projection graphing calculator before today's class begins. The program is given here, along with brief instructions for entering and executing it. If you are not yet comfortable with entering and editing programs, you may want to review the instructions in the Calculator Note "Entering a Program into the Calculator," found in the *The Diver Returns* resources.

Begin the new program listing by pressing the $\boxed{\text{PRGM}}$ key. Use the right arrow to highlight **NEW** and press $\boxed{\text{ENTER}}$ to select **1:Create New**. Enter the program name, **TURNCUBE**, and press $\boxed{\text{ENTER}}$. Enter the programming commands listed here.

PROGRAM: TURNCUBE

Instruction **Explanation**

:-3.7→Xmin The first four commands set the window range. Be sure to use the negative key, $\boxed{\text{(-)}}$, rather than the subtraction key. The arrow is entered with the $\boxed{\text{STO>}}$ key. To enter **Xmin**, press $\boxed{\text{VARS}}$, press $\boxed{\text{ENTER}}$ to select **1:Window**, and then press $\boxed{\text{ENTER}}$ once more to select **Xmin**. At the end of each line in this program, press $\boxed{\text{ENTER}}$ to obtain the colon that signals the beginning of the next command.

:5.7→Xmax Enter this line and the two following lines in the same manner as the first line.

:-2.1→Ymin

:4.1→Ymax

:FnOff Prevents functions that have been entered at the $\boxed{\text{Y=}}$ menu from being displayed. Press $\boxed{\text{VARS}}$, use the right arrow to display the **Y-VARS** menu, press $\boxed{4}$ to select **On/Off**, and press $\boxed{2}$ to select **FnOff**.

:Degree Sets the calculator into degree mode. Press $\boxed{\text{MODE}}$, use the arrow keys to highlight **Degree**, and press $\boxed{\text{ENTER}}$.

:[[1,0,0][1,2,0][3,0,0][3,2,0][1,0,2][1,2,2][3,0,2][3,2,2]]→[A] Stores the three-dimensional coordinates of the eight vertices of the cube as matrix **[A]**. The bracket symbols are $\boxed{\text{2ND}}$ functions to the $\boxed{\times}$ and $\boxed{-}$ keys. The **[A]** cannot be entered using the brackets and the ALPHA character, but must be selected by pressing $\boxed{\text{2ND}}$ [MATRIX] $\boxed{\text{ENTER}}$.

(**Note**: You may find it faster to use the calculator's matrix editor rather than entering the matrix data with the program editor. To do so, press $\boxed{\text{2ND}}$ [MATRIX], use the right arrow to highlight **EDIT**, and press $\boxed{\text{ENTER}}$ to edit matrix **[A]**. Enter **8** and **3** as the matrix dimensions, and then enter the data listed previously. Reenter the program editing mode by pressing $\boxed{\text{PRGM}}$, highlighting **EDIT**, selecting **TURNCUBE**, and then pressing $\boxed{\text{ENTER}}$. Move the cursor to the appropriate line in your program and press $\boxed{\text{2ND}}$ [RCL]. Press $\boxed{\text{2ND}}$ [MATRIX] $\boxed{\text{ENTER}}$ $\boxed{\text{ENTER}}$ to enter the matrix data into your program. Enter →[A] as described previously.)

:[[cos(10),sin(10),0][-sin(10),cos(10),0][0,0,1]]→[C] Stores the matrix for a 10-degree rotation about the *z*-axis as matrix **[C]**. Once again, the **[C]** symbol must be selected from the $\boxed{\text{2ND}}$ [MATRIX] **NAMES** menu and may not be entered using brackets and the letter **C**.

:5→J:4→K:10→L Sets the coordinates of the viewpoint. The colon character is found above the $\boxed{.}$ key.

:4→S Gives the location of the screen as the plane z = 4.

:S/L→R:(S–2)/(L–2)→Q Stores the ratio $\dfrac{z\text{-distance to the screen}}{z\text{-distance to the viewpoint}}$ as **R** for the vertices on the back of the cube and as **Q** for the vertices on the front of the cube.

:{8,2}→dim([B]) Creates a new matrix **[B]**, which will be used to store the two-dimensional coordinates of the projected vertices. Be sure to use braces rather than brackets. The brace symbols are 2ND functions of the parenthesis keys. To enter the **dim** command, press 2ND [MATRIX] use the right arrow to highlight **MATH**, and press 3 to select **dim**. The calculator will add an open parenthesis following **dim**, which you should close at the end of the line. Select the **[B]** symbol from the 2ND [MATRIX] **NAMES** menu.

:For(C,1,37) Begins a loop to draw the cube in each of 37 different orientations. Press PRGM and then 4 to select **For(** from the **CTL** menu.

:For(V,1,4) Begins a loop to calculate the projected coordinates of the vertices on the rear face of the cube and store them in rows 1 through 4 of matrix **[B]**.

:[A](V,1)+R*(J–[A](V,1))→[B](V,1)

:[A](V,2)+R*(K–[A](V,2))→[B](V,2)

:End Closes the V loop. Press PRGM, then press 7 to select **End** from the **CTL** menu.

:For(V,5,8) Begins a loop to calculate the projected coordinates of the vertices on the front face of the cube and store them in rows 5 through 8 of matrix **[B]**.

:[A](V,1)+Q*(J–[A](V,1))→[B](V,1)

:[A](V,2)+Q*(K–[A](V,2))→[B](V,2)

:End Closes the second *V* loop.

:ClrDraw Clears the screen prior to drawing the cube in its new position. Press 2ND [DRAW] and then ENTER to select **1:ClrDraw**.

Shortcut: The next set of 12 lines can be very tedious to enter into the calculator. You may wish to try this shortcut, in which you create a new program containing only a single line, import it into TURNCUBE 12 times, and then edit the lines within TURNCUBE.

1. Press 2ND [QUIT] to exit the TURNCUBE program.

2. Begin a new program, naming it **A** or something similar that will place it at the beginning of your list of programs.
3. Enter the first command beginning with **Line**, below, into program **A**.
4. Press 2ND [QUIT] to exit program **A** .
5. Press PRGM, highlight **EDIT**, and select **TURNCUBE** for editing.
6. Move the cursor to the empty line at the end of your program. Press 2ND [RCL]. Now press PRGM, use the right arrow to highlight **EXEC**, highlight **A** in the list of programs, and press ENTER. Press ENTER once more to insert the contents of program **A** into TURNCUBE.
7. Repeat Step 6 until the **Line** command appears in your program 12 times.
8. Edit the numbers in each line to match those shown below.

:Line([B](1,1),[B](1,2),[B](2,1),[B](2,2)) Press 2ND [DRAW] 2 to select the **Line(** command. **[B]** must be selected from the 2ND [MATRIX] **NAMES** menu.

:Line([B](2,1),[B](2,2),[B](4,1),[B](4,2))

:Line([B](3,1),[B](3,2),[B](4,1),[B](4,2))

:Line([B](1,1),[B](1,2),[B](3,1),[B](3,2))

:Line([B](5,1),[B](5,2),[B](6,1),[B](6,2))

:Line([B](6,1),[B](6,2),[B](8,1),[B](8,2))

:Line([B](7,1),[B](7,2),[B](8,1),[B](8,2))

:Line([B](5,1),[B](5,2),[B](7,1),[B](7,2))

:Line([B](1,1),[B](1,2),[B](5,1),[B](5,2))

:Line([B](2,1),[B](2,2),[B](6,1),[B](6,2))

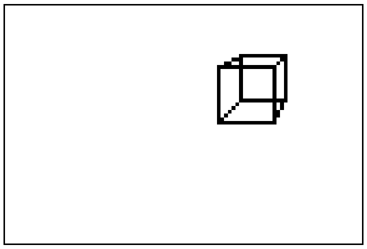
:Line([B](3,1),[B](3,2),[B](7,1),[B](7,2))

:Line([B](4,1),[B](4,2),[B](8,1),[B](8,2))

:[A]*[C]→[A] Applies a 10-degree rotation about the *z*-axis to the vertices in matrix **[A]**. Select each matrix name from the 2ND [MATRIX] **NAMES** menu.

:End Closes the C loop.

Press 2ND [QUIT] to exit program editing mode. Press PRGM, highlight **TURNCUBE**, and press ENTER. Press ENTER again to run the program. (Unlike the screen shown here, you may see the coordinate axes displayed. Subsequent lessons will deal with how to enable or disable that display.)

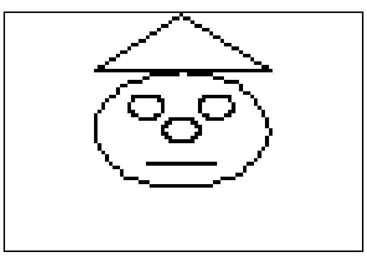


While most Year 4 students have long ago figured out how to draw lines and circles on the screen, they might not know how to do this when the figures are defined in terms of specific coordinates. For the *Picture This!* activity, you may want to assign a particular line segment and circle to be drawn.

Instructions for the **Line, Circle,** and **ClrDraw** commands are provided in the Calculator Note "Drawing by the Numbers." We strongly recommend that you not provide these to students until after the activity is complete, if at all. Finding this information in the calculator's instruction manual is a valuable learning experience, so it is worthwhile to resist the urge simply to tell the class how to perform these tasks. The Calculator Note is primarily for your information, although it is formatted for student use in case you need it for students who were absent for this activity.

When reviewing how to write a program, you may wish to use the Calculator Note "Entering a Program into the Calculator," provided with the *The Diver Returns* resources.

The Calculator Note "Get with the Program and Draw" provides a simple example of how to create a program using some of the basic drawing features that have been discussed. It is provided primarily as a sample for teachers. Although it could be used with the class, students will probably enjoy it more if you let them create a simple picture and talk you through the process of creating the program with which to draw their own creation.



If students are bothered by the appearance of the coordinate axes across their picture, tell them that they will learn to take care of that problem shortly. In fact, this is a good time to begin compiling a posted list of problems that arise as students run their programs over the next several days. Such a list will be a great starting point for the upcoming discussion of what to include in a setup program. Watch for opportunities to add each of these items to the list, although it is not important that every item be included.

- The axes were on/off.

- The screen was not clear of old drawings.
- The viewing window was not set properly.
- Functions entered at $\boxed{\text{Y=}}$ were displayed on top of the picture.
- Points entered with [STAT PLOT] were displayed on top of the picture.
- Circles or squares were distorted.

Because the $\boxed{\text{ZOOM}}$ commands also clear the screen, the **ClrDraw** command in the Calculator Note is not actually necessary. However, it is better at this point to let students see how to insert **ClrDraw** into a program.

Programming Without a Calculator: Students should have written the following programming code for Question 2 of this activity. If you enter the program yourself, add the command **ZStandard** at the beginning of the program. Students will learn to set the viewing window from within their programs in subsequent assignments.

```
Program: LINES
:ClrDraw
:Line(-4,-2,2,-2)
:Line(-4,2,-1,3)
:Line(-1,3,1,5)
:Line(2,-2,4,0)
:Line(1,5,4,4)
:Line(1,5,-2,4)
:Line(-4,2,-2,4)
:Line(4,4,4,0)
:Line(-1,3,2,2)
:Line(-4,2,-4,-2)
:Line(2,2,4,4)
:Line(2,-2,2,2)
```

As students begin to create their own programs, such as in Question 3 of this activity, they will want to safeguard them from accidental erasure; this will become especially important as they work with longer programs that require more than one class period to complete. Such erasures commonly occur when batteries run low or are changed, calculators are dropped, or students in other classes explore what is stored in the calculator. You might share the Calculator Note "Linking Calculators" and encourage students to duplicate their programs on a second calculator daily. Also, you might install TI Connect™ software (free at education.ti.com) on a classroom computer; using this software and a cable, students can save their programs onto the computer.

Learning the Loops: Before students begin this activity, you might have students write a program on the calculator to create a display that alternates between **HI** and **BYE**, as described in the *Teacher's Guide.* Students' initial programs might look like the one shown here, called HIBYE1. You will probably need to review how to use the **Disp** command. It is explained in the Calculator Note "Display and the For Loop," but you might want to hold off on sharing that Calculator Note until you are ready to discuss the syntax of the **For** command.

PROGRAM: HIBYE1

Instruction	Explanation
:Disp "HI"	For **Disp**, press [PRGM], use the right arrow to highlight **I/O** (Input/Output), and press [3]. The space immediately after the **Disp** command is provided by the calculator and does not require entry of a space character. The quotation mark is found above the [+] key.
:Disp "BYE"	
:Disp "HI"	
:Disp "BYE"	
:Disp "HI"	
:Disp "BYE"	

The next program, HIBYE2, illustrates how to write this program using a loop. Although you could begin a new program, this may be a good time to demonstrate how to edit an existing program by inserting new lines. To enter the editing mode, press [PRGM], use the right arrow to highlight **EDIT**, use the down arrow to select your program, and press [ENTER]. To create an empty line in your program in which to insert the **For** command, move the cursor to the beginning of the first line and then press [2ND] [INS] [ENTER]. Move the cursor back to that empty line before entering the **For** command. Remove unwanted **Disp** commands by moving the cursor to the line, pressing [CLEAR] to erase the commands, and then pressing [DEL] to remove the empty line from the program.

Note: In the next program (and in later examples), we have indented the body of the loop for clarity, but the indenting will not appear in the program listing on your calculator screen.

PROGRAM: HIBYE2

Instruction	Explanation
:**For(A,1,3)**	To select **For**, press [PRGM] and then [3].
:**Disp "HI"**	
:**Disp "BYE"**	
:**End**	To select **End**, press [PRGM] and then [4].

The instructions in the Calculator Note "Display and the For Loop" intentionally omit discussion of the optional loop step variable. This will be introduced in the activity *Taking Steps*.

Next in this discussion, as described in the *Teacher's Guide,* have students write a program to display a given set of line segments. The programming code for the suggested program is shown here.

:For(B,2,8)

:Line(1,B,5,B)

:End

The Calculator Note "Setting the Viewing Window from Within a Program" can be used at this point or can be delayed until the later discussion of the setup program.

Learning the Loops: Here is the programming code for Question 1 of this activity.

PROGRAM:LOOP1

Instruction Explanation

:For(T,1,5) **For** is found under the PRGM **CTL** menu.

:Disp "HELLO" **Disp** is found under the PRGM **I/O** menu.

:End **End** is found under the PRGM **CTL** menu.

With the slight modification shown next, you can enter the program for Question 2 into the overhead projection calculator. Students may have had some difficulty with this first encounter with nested loops. The addition of two **Pause** commands allows you to stop the program each time a word is added to the screen display, so that students can predict what will happen next. The calculator only allows program names of up to eight characters, so some condensing of "Fruitloop" will be required.

Program: FRUTLOOP

Instruction Explanation

:For(A,1,5) **For** is found under the PRGM **CTL** menu.

:Disp "LEMON" **Disp** is found under the PRGM **I/O** menu.

:Pause This line should not appear in students' programs but it will be useful in your program. To enter **Pause**, press PRGM, use the down arrow to highlight **8:Pause**, which is below the menu options that are initially visible on the screen, and press ENTER. This command will cause the program to stop each time the word **LEMON** is displayed. The program will remain paused until you press ENTER.

:For(B,1,3)

:Disp "LIME"

:Pause This will cause the program to pause each time the word **LIME** is displayed.

:End End is found under the PRGM **CTL** menu. This command ends the *B* loop.

:End Ends the *A* loop.

Notice that the TI calculator's **End** command does not identify the particular **For** command to which it applies. Each **End** command applies to the most recent **For** command that has not yet been closed with an **End**. Thus, the first **End** command in the program applies to the nearby **For(B,1,3)** command, while the second **End** refers all the way back to the earlier **For(A,1,5)** command.

When you run **FRUTLOOP** with the **Pause** commands added as just shown, the program will stop after each word is added to the screen. Press ENTER to continue the program after any discussion at that point is complete (such as predicting what will happen next).

Here is the programming code for Question 3.

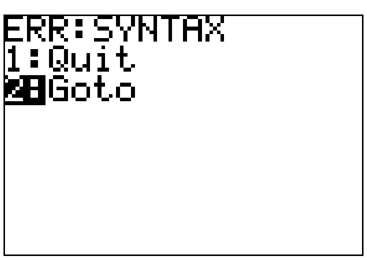
PROGRAM:LOOP2

Instruction	**Explanation**
:For(G,3,10)	**For** is found under the PRGM **CTL** menu.
:Line(G,G,G+3,G+1)	**Line** is found under the 2ND [DRAW] menu.
:End	**End** is found under the PRGM **CTL** menu.

Allow a few students who have entered their programs for Question 4 to display their programs using the overhead calculator projector. If the students' calculators don't plug into the projector, see the instructions in the Calculator Note "Linking Calculators" to transfer their programs to the overhead graphing calculator.

An Animated Shape: Because circles are drawn very slowly on the calculator, they do not lend themselves well to animation. You might want to let students discover this on their own during this activity.

As students encounter error statements when running their programs, encourage them to select the **Goto** option, which will automatically put them back into the program editing mode and will take them directly to the line containing the faulty programming code. A common error will be the use of the subtraction key instead of the negative key.

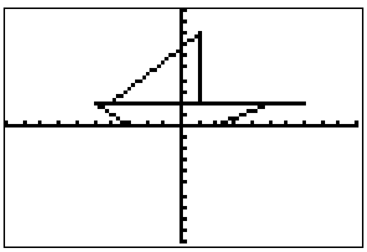


As students complete their work for the day on their programs for *An Animated Shape*, be sure to allow them an opportunity to safeguard their programs by backing them up on another student's calculator. Again, see the Calculator Note *"Linking Calculators."*

Sample programs are provided below, but you will have more fun creating your own. It is a challenge to try to create something interesting with no more than five line segments! Be sure to allow some time for students to demonstrate their programs.

As students work on *An Animated Shape*, be alert for items to add to your posted list of display difficulties when running the programs. (See the previous discussion of this with *Picture This!*) Several such problems are bound to arise. This list will be useful for the discussion of *Movin' On.*

The program that follows, BOAT, is an example of a possible program for *An Animated Shape.* The second program, BOAT2, uses a loop to animate the same boat more simply.



PROGRAM: BOAT

Instruction	**Explanation**
:ZStandard	Sets the viewing window. Press ZOOM 6.
:AxesOn	Ensures that the axes are displayed. (With only five line segments allowed, the *x*-axis was needed as the bottom of the boat!) Press 2ND [FORMAT], move the cursor down to **AxesOn**, and press ENTER.
:Line(−8,0,−5,2)	The next five lines draw the boat.
:Line(−3,2,−15,2)	
:Line(−15,2,−13,0)	
:Line(−9,2,−9,8)	
:Line(−9,8,−14,2)	
:For(T,1,300)	Begins a delay loop.
:End	Ends the delay loop.
:ClrDraw	Clears the screen. Press 2ND [DRAW] 1.
:Line(−3,0,0,2)	Draws the boat in a second position.
:Line(2,2,−10,2)	
:Line(−10,2,−8,0)	

:Line(−4,2,−4,8)

:Line(−4,8,−9,2)

:For(T,1,300)

:End

:ClrDraw

:Line(2,0,5,2) Draws the boat in a third position.

:Line(7,2,−5,2)

:Line(−5,2,−3,0)

:Line(1,2,1,8)

:Line(1,8,−4,2)

:For(T,1,300)

:End

:ClrDraw

:Line(7,0,10,2) Draws the boat in a fourth position.

:Line(12,2,0,2)

:Line(0,2,2,0)

:Line(6,2,6,8)

:Line(6,8,1,2)

The next program animates the boat by using a loop. Notice that the **Line** commands simply add the loop counter *A* to the *x*-coordinates from the **Line** commands in the boat's fourth position. This is an easy way to animate an object moving parallel to one of the axes.

PROGRAM: BOAT2

Instruction	Explanation
:ZStandard	Press $\boxed{\text{ZOOM}}$ $\boxed{6}$.
:AxesOn	Press $\boxed{\text{2ND}}$ [FORMAT], move the cursor down to **AxesOn**, and press $\boxed{\text{ENTER}}$.

:For(A,-20,10)

- **:Line(A+7,0,A+10,2)**
- **:Line(A+12,2,A,2)**
- **:Line(A,2,A+2,0)**
- **:Line(A+6,2,A+6,8)**
- **:Line(A+6,8,A+1,2)**
- **:For(T,1,200)**
- **:End**
- **:ClrDraw** Press $\boxed{\text{2ND}}$ [DRAW] $\boxed{1}$.

:End

Movin' On: Here is programming code for Question 1b of this activity. Setting up a table to keep track of the value of each of the variables for each execution of the loop will be useful. Students are likely to have had difficulty figuring out how to set the variables *T* and *U* equal to the given expressions.

PROGRAM: SEGMENTS

Instruction	Explanation
:−1→Xmin	Students might make a variety of choices for the viewing window settings and for the inclusion of setup commands, other than those shown here. Find **Xmin** and the other window variables in the next several lines by pressing [VARS] [1] and then the appropriate number key to select the desired variable.
:−1→Ymin	
:12→Xmax	
:10→Ymax	
:FnOff	Turns off the display of functions entered at the Y= screen. Press [VARS], use the right arrow to highlight **Y-VARS**, press [4] to select **4:On/Off**, and then press [2] to select **2:FnOff**.
:AxesOn	Turns on the display of the coordinate axes. Press [2ND] [FORMAT], use the down arrow to move the cursor to **AxesOn**, and press [ENTER].
:For(S,1,5)	

:ClrDraw

:S+3→T Sets *T* equal to *S* + 3. The arrow symbol is entered with the [STO>] key.

:S+6→U

:Line(S,S,T,T)

:Line(T,T,U,S)

:Line(U,S,S,S)

:For(A,1,30) Begins a delay loop.

:End Ends the delay loop.

:End Ends the main *S* loop.

One error that can yield unexpected results is to use the same variable for the delay loop counter as is used for the larger loop within which it is nested. If *S* is chosen for the loop counter variable for both loops, the delay loop leaves that variable equal to 31, causing the larger loop to stop after the first time through the loop.

If you have been compiling a posted list of program difficulties as suggested in the *Picture This!* discussion, that list will be helpful in the discussion of

Question 2. Two relevant Calculator Notes are provided. The first, "Suggested Setup Commands," shows how to enter a number of commands that might be useful in a setup program. Consider whether you want to provide these instructions to students at this point. You might prefer to deal at this time only with those commands that students suggest and then add to the list later as further difficulties arise.

The second Calculator Note, "Using a Subroutine," explains how to either run a setup subprogram from within a program or import the instructions from a setup program into the listing of another program. The first of these two techniques provides a nice exposure to the use of subprograms; the second results in handier stand-alone programs and is also a useful tool for importing a complex or lengthy command that will be used many times within a program.

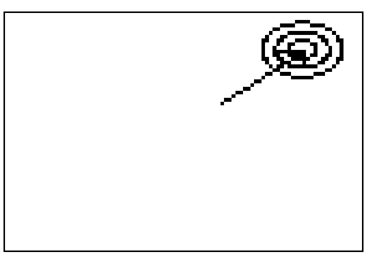
Some Back and Forth: One option for the programming code for Question 2 is shown here. See the Calculator Notes "Suggested Setup Commands" if you need help entering the setup commands. Of course, many other reasonable choices of setup commands and viewing window settings are possible.

Instruction	**Explanation**
:−1→Xmin	This and the next three instructions set the viewing window. **Xmin** and so on are found on the VARS **Window** menu.
:4→Xmax	
:−1→Ymin	
:4→Ymax	
:AxesOn	The activity does not require that the axes be displayed, so this could be omitted. Press 2ND [FORMAT].
:FnOff	This is found under VARS **Y-VARS**.
:PlotsOff	Press 2ND [STAT PLOT] 4.
:For(A,1,4)	
:ClrDraw	Press 2ND [DRAW] 1.
:Line(1,1,2,2)	
:For(B,1,100)	Be careful not to use *A* again as the loop counter here.
:End	
:ClrDraw	
:Line(2,2,3,3)	
:For(B,1,100)	
:End	
:End	

Arrow: This programming code is one possible solution for this activity. The choice of setup commands and viewing window settings could vary considerably.

PROGRAM:ARROW

Instruction	**Explanation**
:ZStandard	Pressing $\boxed{\text{ZOOM}}$ $\boxed{6}$ is the quickest way to set **Xmax** and **Ymax** to **10**.
:-3→Xmin	**Xmin** is found on the $\boxed{\text{VARS}}$ **Window** menu.
:-3→Ymin	
:AxesOff	This is found under $\boxed{\text{2ND}}$ [FORMAT].
:FnOff	This is found under $\boxed{\text{VARS}}$ **Y-VARS**.
:PlotsOff	Press $\boxed{\text{2ND}}$ [STAT PLOT]y.
:For(A,1,8)	
:ClrDraw	
:A–3→B	
:A–1→C	
:Line(B,B,A,A)	
:Line(C,A,A,A)	
:Line(A,C,A,A)	
:For(D,1,30)	Be careful not to use *A* again as the loop counter for the delay loop.
:End	
:End	



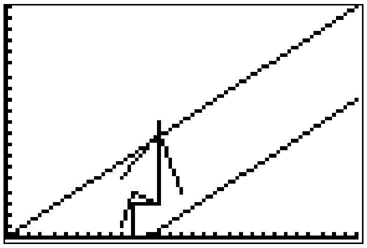
As students strive to find ways to improve the *Arrow* program, someone will undoubtedly try giving it a circular target to hit. The result is not very satisfactory, because circles take so long to draw that they slow the animation down to a crawl. The Calculator Note “Storing and Recalling Pictures” describes how to save the target as a picture so that it can be instantly recalled. It is worth taking some time to deal with this now, because this technique will be very useful as students create their own animations for *POW 5: An Animated POW*. However, because the instructions use a modification of the *Arrow* program as an example, do not provide them to students until after they have completed their own improvements to the program.

Sunrise: In preparation for this activity, remind students that they will want to use the ZOOM option, **ZSquare**, to keep their circles from being distorted. Because **ZSquare** adjusts the window variables, it should be used only after all other commands that affect the window range have been entered.

Move That Line!: The programming code for Question 3 (using a loop) should look similar to what's shown here, although the exact numbers will vary depending on the line segment and translation that were chosen.

Instruction	**Explanation**
:−1→Xmin	A number of variations to the setup commands and viewing window settings are possible. **Xmin** is found under the VARS **Window** menu.
:21→Xmax	
:−20→Ymin	
:3→Ymax	
:FnOff	This is found under VARS **Y-VARS**.
:PlotsOff	Press 2ND [STAT PLOT] 4.
:AxesOn	This is found under 2ND [FORMAT].
:3→P	
:2→Q	
:5→R	
:1→S	
:For(N,1,6)	
:ClrDraw	
:Line(P,Q,R,S)	
:For(A,1,50)	This starts a delay loop. Be careful not to use *P, Q, R, S,* or *N* as the loop counter.
:End	
:P+3→P	
:R+3→R	
:Q-4→Q	
:S-4→S	
:End	

The next program provides a sample of the type of program that might be created for Question 4. If a student comes up with a working program, it will be more interesting to use that program for illustration than the one provided here. (The program shown here uses slightly more than the four or five line segments suggested in Question 4.)



Instruction **Explanation**

:0→Xmin

:30→Xmax

:0→Ymin

:20→Ymax

:FnOff

:PlotsOff

:20→A:19→B:14→C:13→D The colon separating commands is obtained by pressing $\boxed{\text{ALPHA}}$ $\boxed{.}$.

:11→E:10→F:25→G:26→H

:28→I:30→J:15→K

:For(X,1,6)

:ClrDraw

:Line(30,20,0,0)

:Line(30,12,12,0)

:Line(I,A,I,D)

:Line(I,B,G,K)

:Line(I,B,J,C)

:Line(I,D,H,C)

:Line(H,C,G,E)

:Line(I,D,H,D)

:Line(H,D,H,F)

:A-2→A:B-2→B:C-2→C

:D-2→D:E-2→E:F-2→F

:K-2→K:G-3→G:H-3→H

:I-3→I:J-3→J

:For(Y,1,50)

:End

:End

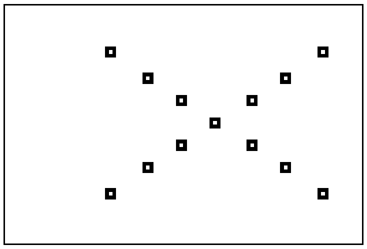
Double Dotting: Although students were not asked to create programming code in this activity, two issues are likely to have caused confusion. The first is how to draw dots with the calculator. The needed command, **Pt-On**, is found by pressing 2ND [DRAW] and then using the right arrow key to highlight **POINTS**. **Pt-On** must be followed by the *x*- and *y*-coordinates of the point, within parentheses and separated by commas. A third argument can follow the *y*-coordinate to define the style of the dot. The available styles are: 1 = dot, 2 = box, 3 = +.

The second issue that may cause confusion is how to increase or decrease the variables. The solution is to add a value to (or subtract it from) the variable and then store it to the same variable name, such as **A+1→A**.

Possible programming code for Questions 1 and 2 is shown here. The screen display is using point style 2.

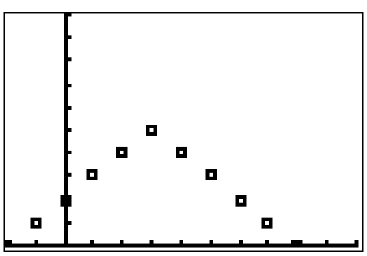
PROGRAM: DOTS

Instruction	Explanation
:ZStandard	Pressing ZOOM 6 will quickly set **Xmax** and **Xmin** to 10.
:0→Xmin	**Xmin** is found under the VARS **Window** menu.
:0→Ymin	
:FnOff	This is found under VARS Y-VARS.
:PlotsOff	Press 2ND [STAT PLOT] 4.
:3→A	
:2→B	
:9→C	
:2→D	
:For(P,1,7)	
:Pt-On(A,B,2)	Press 2ND [DRAW], use the right arrow to highlight **POINTS**, and then press 1 to select **1:Pt-On(**.
:Pt-On(C,D,2)	
:A+1→A	
:B+1→B	
:D+1→D	
:C-1→C	
:End	



PROGRAM: MOREDOTS

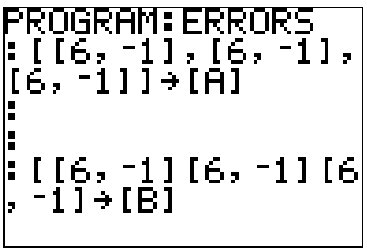
Instruction	Explanation
:ZStandard	
:−2→Xmin	
:0→Ymin	
:FnOff	
:PlotsOff	
:AxesOn	This is found under $\boxed{\text{2ND}}$ [FORMAT].
:3→A	
:5→B	
:3→C	
:Pt-On(A,B,2)	
:For(P,1,5)	
:A-1→A	
:B-1→B	
:C+1→C	
:Pt-On(A,B,2)	
:Pt-On(C,B,2)	
:End	



Memories of Matrices: After a little experimentation, most students will remember from their work in Year 3 how to use the matrix editor and how to perform matrix addition and multiplication on the calculator. They may have considerable difficulty, however, in discovering from the calculator manual how to define a matrix or address a matrix element from within a program. The instructions in the Calculator Note *"Matrices"* will clarify these issues.

As students experiment, be on the lookout for several common errors. Matrix names must be selected from the 2ND [MATRIX] **NAMES** menu; they cannot be typed in using the bracket keys and the ALPHA characters. One way to detect this error is to move the cursor along the matrix name. If it has been typed in, you can position the cursor on any of the three symbols (the two brackets and the letter). If it has been entered properly from the 2ND [MATRIX] **NAMES** menu, you cannot position the cursor individually on these three elements of the matrix name.

Two additional common errors when defining a matrix in a program involve entering commas between the brackets separating rows of a matrix (as in the first example in the illustration here) or failing to enter all of the necessary brackets. (The second example in the illustration here is missing a needed bracket just before the arrow symbol. The last row of the matrix was closed with a bracket, but the entire list of matrix elements must be enclosed in brackets as well.)



Though a program can use matrices that were defined externally (not from within the program), discourage students from falling into this habit. For a program to run reliably, it should not depend on someone manually entering the matrices each time it is used.

Comin' Round Again (and Again...): During the discussion of this activity, remind students of the need to be careful about closing the parentheses the calculator opens for each trigonometric function. **5cos(60–3sin(60**, for example, will not yield the desired result, because everything after the first parenthesis will be considered to be enclosed within the parentheses.

Taking Steps: This activity adds one more parameter to the **For** command: an increment or step value. The step value can be positive or negative and is enclosed within the parentheses for the **For** command, following the final value. The programming code for Question 1, shown here, provides an example of the syntax.

```
:For(C,2,11,3)
    :Disp C
:End
```

The programming code for Questions 2 and 3 is very similar to that for Question 1. Here is one possibility for Question 4.

```
:For(B,5,1,–1)
    :Disp B
:End
```

Swing That Line!: Students may wish to create their program for this activity by copying and then modifying their program from *Move That Line!* For instructions for copying a program, see the instructions under *Copying a Program* in the Calculator Note "Using A Subroutine." Summarized briefly, these are the steps.

- Open a new program.
- From within the new program, press $\boxed{\text{2ND}}$ [RCL].
- Press PRGM and use the right arrow to highlight **EXEC**.
- Use the down arrow to highlight the name of the old program and press $\boxed{\text{ENTER}}$.
- Press $\boxed{\text{ENTER}}$ again to import the commands from the old program into the new.

The programming code given here follows the outline of the plain-language program presented discussion of this activity in the *Teacher's Guide*.

Instruction	Explanation
:ZStandard	Press $\boxed{\text{ZOOM}}$ $\boxed{6}$.
:AxesOn	This is found under $\boxed{\text{2ND}}$ [FORMAT].
:FnOff	Press $\boxed{\text{VARS}}$ **Y-VARS** $\boxed{4}$ $\boxed{2}$.
:PlotsOff	Press $\boxed{\text{2ND}}$ [STAT PLOT] $\boxed{4}$.
:Degree	Press $\boxed{\text{MODE}}$ and select **Degree**.
:[[3,2][5,1]]→[A]	Select **[A]** under $\boxed{\text{2ND}}$ [MATRIX] **NAMES**.

```
:[[cos(15),sin(15)][-sin(15),cos(15)]]→[B]
:For(J,1,11)
    :ClrDraw
    :Line([A](1,1),[A](1,2),[A](2,1),[A](2,2))
    :For(D,1,30)
    :End
    :[A][B]→[A]
:End
```

If students choose to enter the data for matrix **[B]** by using the calculator's matrix editor (as described in the Calculator Note "Matrices"), they will need to be sure they are in degree mode first. They will also find that when they import the matrix into the calculator, the sine and cosine functions will not be displayed as here. Instead, the numerical values of those functions will be displayed.

As students are testing their programs, watch for anyone who has trouble because the calculator is in radian mode instead of degree mode. Use that opportunity to prompt students to suggest adding **Degree** to their setup programs. If this does not arise, it should be mentioned anyway.

What's Going On Here?: The programming code for this activity is presented here. The matrices can be created most easily by using the calculator's matrix editor and then importing them into the program, as described in the Calculator Note "Matrices."

PROGRAM:MYSTERY

Instruction	Explanation
:ZStandard	A number of variations are possible for the setup commands shown here.

:ZSquare

:AxesOn

:Degree

:FnOff

:PlotsOff

:ClrDraw

:[[2,2][5,2][2,6][5,6][3.5,9]]→[A]

:[[cos(30),sin(30)][-sin(30),cos(30)]]→[B]

:For(C,1,12)

The **Line** commands can be quite tedious to enter. Students may find it easier to create a new program containing only the first line and then to import that program into their MYSTERY program five times, using the 2ND [RCL] feature. They can make appropriate changes to those imported lines.

> **:Line([A](1,1),[A](1,2),[A](2,1),[A](2,2))**
>
> **:Line([A](2,1),[A](2,2),[A](4,1),[A](4,2))**
>
> **:Line([A](4,1),[A](4,2),[A](5,1),[A](5,2))**
>
> **:Line([A](1,1),[A](1,2),[A](3,1),[A](3,2))**
>
> **:Line([A](3,1),[A](3,2),[A](5,1),[A](5,2))**
>
> **:[A][B]→[A]**

:End

POW 5: An Animated POW: In doing this POW, students often get quite creative and invariably choose projects that push the limits of their programming abilities. Even if their programs do not usually include projections (introduced over the next several days), they are a tremendous learning experience.

Some time is scheduled later in the unit for students to work on their programs in class. However, once they begin to program, you will find it useful to provide several class periods for this if you can afford the time. In any case, you will need to make provisions for students to take calculators home overnight. They will find that programming an original creation like this is very time consuming. But generally, they get caught up enough in the project that they do not resent the time requirements. Make the project worth an appropriate number of grade points, probably more than a typical POW.

You should also remind students now, as well as several times throughout the remainder of this unit, to back up their POW programs to another calculator or computer after each session of work. It is very frustrating to lose several evenings' work because the batteries ran low or someone dropped the calculator.

As students begin to work on the POW over the next week, the question is bound to arise of how to redraw background objects more quickly after the screen is cleared during animation, especially if circles are involved. The instructions in the Calculator Note "Storing and Recalling Pictures" will be useful in overcoming this problem. Remind students, however, that it is much more effective if their program first draws and stores the background picture, rather than having to draw it manually and recall it with the program. They should strive for programs that stand alone as much as possible, so needed bits and pieces won't get lost or erased.

Another Mystery: The programming code for this activity is very similar to that for *What's Going On Here?,* presented earlier in this guide.

Find Those Corners!: The supplemental problem *Let the Calculator Do It!* asks students to create a calculator program that will perform the calculations for this activity, The program given as a solution to that supplemental problem may be useful for quickly checking students' progress in *Find Those Corners!* Students will be seeking reassurance as they complete portions of the complex calculations. But checking their work quickly can be a problem because each group will be working with a different viewpoint and probably will be working with the cube's vertices in a different order.

The Turning Cube Outline: After you discuss this activity, if you choose to have conclude the unit by having students explain the program TURNCUBE, you can photocopy that program, which is provided in the Calculator Note "The TURNCUBE Program." The following description will help guide the discussion. Most of this is also included in this calculator guide for *Picture This!*, along with instructions for entering each command.

PROGRAM:TURNCUBE

Instruction Explanation

:–3.7→Xmin These four commands set the window range.

:5.7→Xmax

:–2.1→Ymin

:4.1→Ymax

:FnOff Prevents functions that have been entered at the $\boxed{\text{Y=}}$ menu from being displayed.

:Degree Sets the calculator into degree mode.

:[[1,0,0][1,2,0][3,0,0][3,2,0][1,0,2][1,2,2][3,0,2][3,2,2]]→[A] Stores the three-dimensional coordinates of the eight vertices of the cube as matrix **[A]**. This matrix will represent the cube's current position.

:[[cos(10),sin(10),0][-sin(10),cos(10),0][0,0,1]]→[C] Stores the matrix for a 10-degree rotation about the *z*-axis as matrix **[C]**.

:5→J:4→K:10→L Sets the coordinates of the viewpoint.

:4→S Gives the location of the screen as the plane $z = 4$.

:S/L→R:(S–2)/(L–2)→Q Stores the ratio $\frac{z\text{-distance to the screen}}{z\text{-distance to the viewpoint}}$ as **R** for the vertices on the back of the cube and as **Q** for the vertices on the front of the cube. The ratio **R** appears so much simpler because the *z*-coordinate for these vertices is zero. (The ratio could be written as **(S–0)/(L–0)**.)

:{8,2}→dim([B]) Creates matrix **[B]**, which will be used to store the two-dimensional coordinates of the projected vertices. It is necessary to dimension the matrix here because the matrix elements were not defined previously, as were those of matrix **[A]**.

:For(C,1,37) Begins a loop to draw the cube in each of 37 different orientations, the original position and 36 rotations in 10-degree increments, for a full 360-degree rotation.

:For(V,1,4) Begins a loop to calculate the projected coordinates of the vertices on the rear face of the cube and store them in rows 1 through 4 of matrix **[B]**.

:[A](V,1)+R*(J-[A](V,1))→[B](V,1) Calculates the *x*-coordinate of the projection of one of the four vertices on the rear of the cube and stores it in column 1 of matrix **[B]**.

:[A]V,2)+R*(K-[A](V,2))→[B](V,2) Calculates the *y*-coordinate of the projection of one of the four vertices on the rear of the cube and stores it in column 2 of matrix **[B]**.

:End Closes the *V* loop.

:For(V,5,8) Begins a loop to calculate the projected coordinates of the vertices on the front face of the cube and store them in rows 5 through 8 of matrix **[B]**.

:[A](V,1)+Q*(J-[A](V,1))→[B](V,1) Calculates the *x*-coordinate of the projection of one of the four vertices on the front of the cube and stores it in column 1 of matrix **[B]**.

:[A](V,2)+Q*(K-[A](V,2))→[B](V,2) Calculates the *y*-coordinate of the projection of one of the four vertices on the front of the cube and stores it in column 2 of matrix **[B]**.

:End Closes the second *V* loop.

:ClrDraw Clears the screen prior to drawing the cube in its new position.

:Line([B](1,1),[B](1,2),[B](2,1),[B](2,2)) These commands draw the 12 line segments that make up the projection of the edges of the cube.

:Line([B](2,1),[B](2,2),[B](4,1),[B](4,2))

:Line([B](3,1),[B](3,2),[B](4,1),[B](4,2))

:Line([B](1,1),[B](1,2),[B](3,1),[B](3,2))

:Line([B](5,1),[B](5,2),[B](6,1),[B](6,2))

:Line([B](6,1),[B](6,2),[B](8,1),[B](8,2))

:Line([B](7,1),[B](7,2),[B](8,1),[B](8,2))

:Line([B](5,1),[B](5,2),[B](7,1),[B](7,2))

:Line([B](1,1),[B](1,2),[B](5,1),[B](5,2))

:Line([B](2,1),[B](2,2),[B](6,1),[B](6,2))

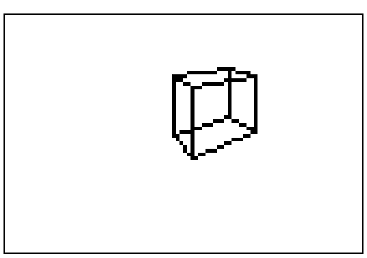
:Line([B](3,1),[B](3,2),[B](7,1),[B](7,2))

:Line([B](4,1),[B](4,2),[B](8,1),[B](8,2))

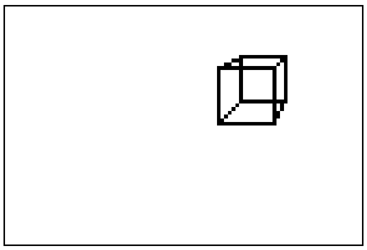
:[A]*[C]→[A] Applies a 10-degree rotation about the *z*-axis to the vertices in matrix **[A]**, so that the cube will be drawn in a new position the next time.

:End Closes the *C* loop. No delay loop is necessary because the time required for the calculations provides a natural delay.

An interesting extension that some students may choose to pursue as an improvement to the program is to cause the cube to rotate about the *y*-axis rather than about the *z*-axis. The result is quite impressive, with the size of the cube changing as it gets closer or farther away, as in the screens shown here.



Rotation about the *y*-axis requires only a few changes to the program. The rotation matrix must be replaced with the matrix for rotation about the *y*-axis, as described in the *Teacher's Guide* for the activity *Rotation Matrix in Three Dimensions*. Because a face of the cube does not remain parallel to the *xy*-plane, the ratio *R* cannot be calculated for four vertices at once. And finally, because the *z*-coordinates of the cube's vertices are changing as the cube moves, the calculation of *R* must be moved inside of the inner **For** loop, where it will be performed for each vertex each time the cube is drawn.



The next program incorporates these modifications.

```
:–3.7→Xmin
:5.7→Xmax
:–2.1→Ymin
:4.1→Ymax
:FnOff
:Degree
:[[1,0,0][1,2,0][3,0,0][3,2,0][1,0,2][1,2,2][3,0,2][3,2,2]]→[A]
:[[cos(10),0,sin(10)][0,1,0][—sin(10),0,cos(10)]]→[C]
:5→J:4→K:10→L
:4→S
:{8,2}→dim([B])
:For(C,1,37)
    :For(V,1,8)
    :(S–[A](V,3))/(L–[A](V,3))→R
    :[A](V,1)+R*(J– [A](V,1))→[B](V,1)
    :[A](V,2)+R*(K–[A](V,2))→[B](V,2)
    :End
    :ClrDraw
    :Line([B](1,1),[B](1,2),[B](2,1),[B](2,2))
    :Line([B](2,1),[B](2,2),[B](4,1),[B](4,2))
    :Line([B](3,1),[B](3,2),[B](4,1),[B](4,2))
    :Line([B](1,1),[B](1,2),[B](3,1),[B](3,2))
    :Line([B](5,1),[B](5,2),[B](6,1),[B](6,2))
```

```
    :Line([B](6,1),[B](6,2),[B](8,1),[B](8,2))
    :Line([B](7,1),[B](7,2),[B](8,1),[B](8,2))
    :Line([B](5,1),[B](5,2),[B](7,1),[B](7,2))
    :Line([B](1,1),[B](1,2),[B](5,1),[B](5,2))
    :Line([B](2,1),[B](2,2),[B](6,1),[B](6,2))
    :Line([B](3,1),[B](3,2),[B](7,1),[B](7,2))
    :Line([B](4,1),[B](4,2),[B](8,1),[B](8,2))
    :[A]*[C]→[A]
:End
```

"An Animated POW" Write-up: As students prepare to complete *"An Animated POW" Write-up,* if you have the TI Connect (available free at education.ti.com) and a cable for linking the calculator to a computer, you may wish to provide students with an opportunity to print listings of their programs from the computer. Preparing a written copy of a lengthy program by hand can be very tedious.

In-Class Assessment: If you have more than one class of Year 4 students, or if you have students doing a make-up of the exam, you will want to consider carefully how to handle clearing solutions to the *In-Class Assessment* from the calculators between classes. The quickest and surest way is to reset the calculator's memory. The difficulty is that this will erase all programs from the calculator, including student work from *POW 5: An Animated POW.* The best solution is to borrow a set of calculators that have no critical programs stored from another teacher for the day. To quickly reset the memory of a the calculator, press [2ND] [MEM] [5] [1] [2] and then adjust the contrast with [2ND] [↑].

Supplemental Problems

Loopy Arithmetic: One possible program is presented here for each of the two problems in *Loopy Arithmetic*. FCTORIAL will compute the factorial of a user-supplied number. FIBONACI displays the first 40 Fibonacci numbers.

PROGRAM: FCTORIAL

Instruction	Explanation
:Input "NUMBER?",N	Prompts the user to enter a number and stores the number as variable *N*. Input is found under the [PRGM] **I/O** menu. The question mark is an ALPHA key above the negative key ([(−)]).
:1→F	
:For(A,1,N)	**For** is found under the [PRGM] **CTL** menu.

:F*A→F Up to the number that was input, computes the factorial of each integer in sequence by multiplying it by the factorial of the previous integer.

:End **End** is found under the $\boxed{\text{PRGM}}$ **CTL** menu.

:Disp F Displays the final value of *F*. **Disp** is found under the $\boxed{\text{PRGM}}$ **I/O** menu.

PROGRAM: FIBONACI

Instruction **Explanation**

:0→A **A** and **B** will be used to keep track of the two Fibonacci numbers that precede the one being calculated.

:1→B

:For(D,1,40) **For** is found under the $\boxed{\text{PRGM}}$ **CTL** menu.

:Disp B Displays each Fibonacci number. **Disp** is found under the $\boxed{\text{PRGM}}$ **I/O** menu.

:A+B→C Adds the two previous Fibonacci numbers in order to find the next. Temporarily stores it as **C**.

:B→A Moves the number currently in **B** into **A**, to reflect that it is now the Fibonacci number two terms previous to the next one to be calculated.

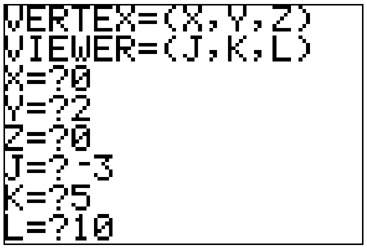
:C→B Moves the new Fibonacci number in **C** into **B**, reflecting that it will be the Fibonacci number immediately previous to the next one to be calculated.

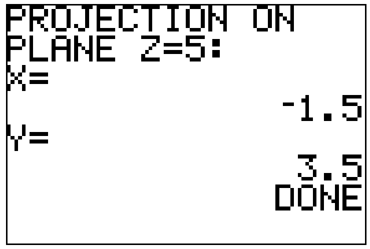
:Disp "PRESS $\boxed{\textbf{ENTER}}$" Prompts the user to press the $\boxed{\text{ENTER}}$ key to calculate the next Fibonacci number.

:Pause Causes the program to pause after the display of each new number, so that the numbers don't scroll from the screen too fast to be read. (Alternately, a delay loop could be used here.) To enter **Pause**, press $\boxed{\text{PRGM}}$ and use the down arrow to move the cursor to **8:Pause** under the **CTL** menu. It will be below the items that are initially displayed on the screen. Press $\boxed{\text{ENTER}}$.

:End Press PRGM to find this under the **CTL** menu.

Let the Calculator Do It!: The next program, PROJECTN, is one possible solution to the supplemental problem *Let the Calculator Do It!* It will calculate the coordinates of the projection of any point onto the plane $z = 5$, as seen from any viewpoint. (Be careful. The program does not check to see if the point and the viewpoint are on opposite sides of the plane $z = 5$.) You may find the program useful for quickly checking student work in progress during the activity *Find Those Corners!*





PROGRAM: PROJECTN

Instruction **Explanation**

:ClrHome Clears the home screen. Press PRGM, use the right arrow to highlight **I/O**, use the down arrow to scroll down to **8:ClrHome**, and press ENTER.

:Disp "VERTEX=(X,Y,Z)" Identifies the variables for the user. **Disp** is under the PRGM **I/O** menu. To enter the equal sign, press 2ND [TEST] 1.

:Disp "VIEWER=(J,K,L)"

:Prompt X:Prompt Y:Prompt Z Prompts the user to enter a value for each variable by displaying the variable name, followed by **=?**, and then assigns the value entered by the user to that variable. **Prompt** is under the PRGM **I/O** menu. The colon is the ALPHA function above the decimal key (.).

:Prompt J:Prompt K:Prompt L

:(5–Z)/(L–Z)→R Calculates the ratio.

:X+R(J–X)→A Calculates the *x*-coordinate of the projection.

:Y+R(K–Y) →B Calculates the *y*-coordinate of the projection.

:ClrHome

:Disp "PROJECTION ON" The space between words is the ALPHA character above the zero key.

:Disp "PLANE Z=5:" Press 2ND [TEST] 1 for the equal sign.

:Disp "X=",A Displays the coordinates of the projected point.

:Disp "Y=",B

Drawing by the Numbers

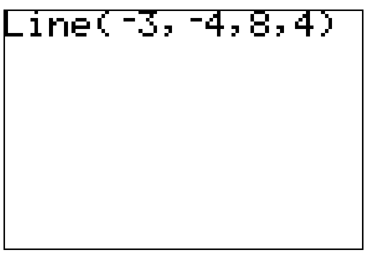
You may already know how to draw line segments and circles on the calculator screen. One way to draw a line is to press GRAPH to go to the graphing screen and then press 2ND [DRAW] to view a list of available drawing commands. Press 2 to select **2:Line(**. Move your cursor to the position for one endpoint of your line segment and press ENTER. Then move it to the position of the other endpoint and press ENTER again.

To draw a line within a program, you tell the program the pair of coordinates to use as the endpoints. To practice this technique, you will need to be on the home screen instead of the graphics screen.

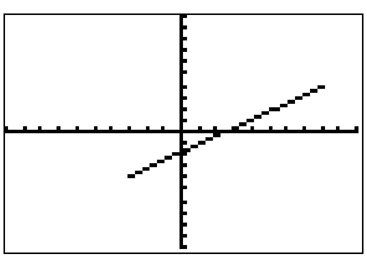
Drawing Line Segments

Press 2ND [QUIT] to return to the home screen. As before, press 2ND [DRAW] and then press 2 to select the **Line** command. The results will be different from what happens when this command is selected from the graphics screen. Instead of allowing you to locate the line segment with the cursor, the **Line** command will be displayed on your home screen.

Within the parentheses that have been opened, enter the *x*- and *y*-coordinates of one endpoint, separated by a comma. (The comma key is above the 7 key.) Put another comma after the *y*-coordinate and then enter the coordinates of the other endpoint in the same way. Both sets of coordinates will be enclosed within the same set of parentheses.



Close the parentheses and press ENTER. The calculator will shift to the graphing screen, where you should see the line segment you defined. If not, you probably need to adjust the window range to include your line segment. Press WINDOW and enter values for **Xmin, Xmax, Ymin,** and **Ymax** to define the coordinates of the left, right, bottom, and top edges of the screen, respectively.



Drawing Circles

Circles are drawn similarly to line segments. Once again, press 2ND [QUIT] to return to the home screen and then 2ND [DRAW] to display the draw menu. Use the down arrow to move down and highlight **9:Circle(**. Press ENTER to copy the **Circle** command to the home screen.

Within the parentheses, enter the *x*- and *y*-coordinates of the center of the circle, followed by the radius, all separated by commas. Close the parentheses and press ENTER to view the circle being drawn.

Depending on the viewing window you have selected, your circle may appear more like an ellipse. This is because your viewing window may cause the graph to be stretched in one direction. This cannot be fixed by assigning the same ranges to the *x*- and *y*-axes at the WINDOW screen—because the screen itself is not square, a greater range must be assigned to the *x*-axis than to the *y*-axis to prevent distortion. The simplest way to correct this is to press ZOOM and select **5:ZSquare**. This will automatically widen your selected window settings in one direction to provide a distortion-free graph. When you execute this command, however, any previous drawings will be erased. To bring the previous drawing command back immediately, press 2ND [QUIT] ENTER. Use the 2ND [ENTRY] command to recall even older instructions.

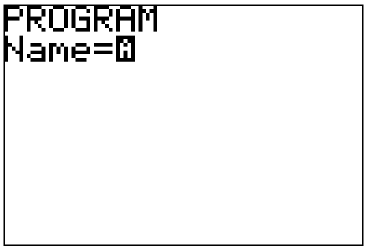
Clearing the Graphics Screen

While pressing the CLEAR key works well for clearing the home screen, it has no effect on the graphing screen. To clear the graphing screen, press 2ND [DRAW] and select **1:ClrDraw**. If you do this from the graphing screen, your drawing will be immediately cleared. If you do this from the home, screen, the **ClrDraw** command will be displayed, and you will need to press ENTER to execute it.

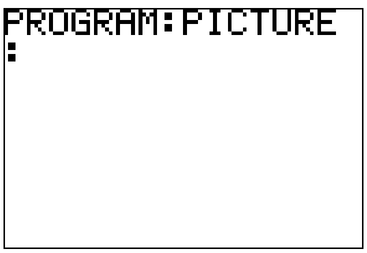
Get with the Program and Draw!

The Calculator Note “Drawing by the Numbers” explains how to draw lines and circles. This Calculator Note tells you how to incorporate those features into a program.

To begin a new program, press the PRGM key, use the right arrow to highlight **NEW**, and press ENTER. Enter a name for the program that is no more than eight characters long. We will call this one PICTURE.



After entering the program name, press ENTER to obtain the screen shown here. Each new line of the program will be preceded with a colon, which the calculator will supply whenever you press ENTER at the end of a line. Enter the programming code shown here.



PROGRAM: PICTURE

Instruction	**Explanation**
:ClrDraw	Press 2ND [DRAW] and then press ENTER to select **1:ClrDraw**. Press ENTER again to move to the next line (and at the end of each line that follows).
:ZStandard	Sets the viewing window. Press ZOOM and then 6 to select **6:ZStandard**.
:Line(−5,5,5,5)	Press 2ND [DRAW] and then 2 to select **2:Line(**. Be sure to use the (−) key, and not the subtraction key, for the negative sign.
:Line(−5,5,0,10)	Don’t forget to press ENTER at the end of this statement. Because of the length of this statement, your cursor will automatically move to the next line. You are not ready to enter the following statement, however, until you press ENTER and can see the colon at the beginning of the new line.
:Line(0,10,5,5)	

:Line(−2,−3,2,−3)

:Circle(0,0,5) Press 2ND [DRAW], use the down arrow to highlight **9:Circle(**, and press ENTER.

:Circle(0,0,1)

:Circle(−2,2,1)

:Circle(2,2,1)

Press 2ND [QUIT] to exit the program editing mode. Press PRGM, highlight **PICTURE** (or your program name) under the **EXEC** menu, and press ENTER. When **prgmPICTURE** appears on the home screen, press ENTER again to run the program.

Linking Calculators

Calculators can be linked with another calculator in order to exchange programs and files. Consider taking a few minutes each day to link calculators with another student and provide each other with backup copies of the programs you're working on.

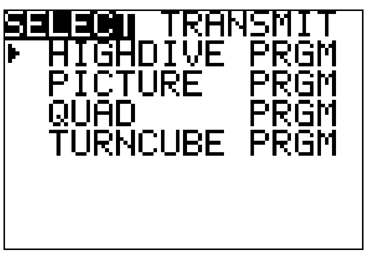
These instructions describe how to transmit a program from one calculator to another.

1. To link calculators, first plug one end of the link cable that comes with the calculators into each calculator. When you feel it snap into place, push a little more. Failure to push the plug fully into place is the most common reason for failure to link.

2. On the receiving calculator, press $\boxed{\text{2ND}}$ [LINK]. Use the right arrow to highlight **RECEIVE** and press $\boxed{\text{ENTER}}$. The calculator display will tell you that it is **Waiting**.

3. On the sending calculator, press $\boxed{\text{2ND}}$ [LINK] and use the down arrow to highlight **2:All–…**. Press $\boxed{\text{ENTER}}$ to bring up a list of your programs and files.

4. Use the down arrow to move the arrow cursor to the name of a program to be transmitted, and press $\boxed{\text{ENTER}}$. A square will appear before the program name, marking it as selected. Select any additional programs to be transmitted in the same way.



5. When you have selected all the programs you wish to transmit, use the right arrow to highlight **TRANSMIT** and press $\boxed{\text{ENTER}}$.

Display and the For Loop

The Display Command

Text or numerical values can be displayed on the home screen from within a program using the **Disp** command. To enter this command, press the $\boxed{\text{PRGM}}$ key while entering or editing a program and then use the right arrow to highlight **I/O** (Input/Output). Press $\boxed{3}$ to select **3:Disp**. (You cannot find this menu by pressing $\boxed{\text{PRGM}}$ from the home screen.)

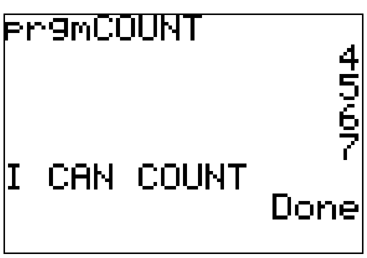
The **Disp** command must be followed by a number, variable, or string of text to be displayed. Notice each of these examples.

Instruction	Explanation
:Disp 7	Displays the number 7.
:Disp R	Displays the current numerical value of the variable *R*.
:Disp "MY NAME"	Displays the characters **MY NAME**. The space between the two words is the ALPHA character above the $\boxed{0}$. Quotation marks must be used around the characters to be displayed.
:Disp "ANSWER=",T	Displays the characters **ANSWER=**, followed by the value of the variable *T*. The equal sign is found by pressing $\boxed{\text{2ND}}$ [TEST].

The For Loop

The **For** and **End** commands indicate the beginning and end of a loop. You can find both commands under the **EXEC** menu by pressing $\boxed{\text{PRGM}}$ while editing or entering a program. Press $\boxed{4}$ to select **For** or $\boxed{7}$ to select **End**.

The **For** command must be followed by a set of parentheses containing a variable to use as a counter, an initial value for that counter, and a final value for the counter, all separated by commas. The example here displays a count from 4 to 7. Notice that once the loop is completely finished, the program continues with the next line after the **End** command.



:For(A,4,7)

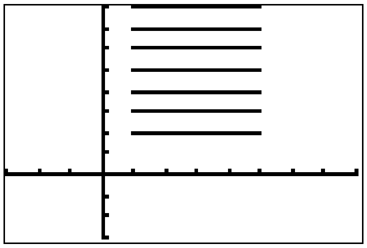
:Disp A

:End

:Disp "I CAN COUNT"

Setting the Viewing Window from Within a Program

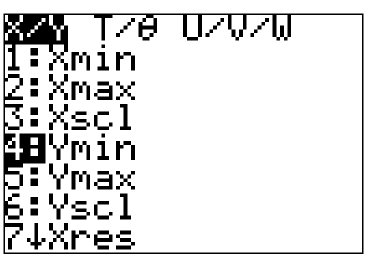
You have probably just created a program to draw the display shown here. To get your display to look like this, you probably had to adjust the viewing window at the WINDOW screen. These instructions will tell you how to adjust the viewing window from within a program, so that you will not have to adjust it manually every time you run your program.



The quickest way to set a viewing window from within a program is to insert a ZOOM command into your program listing. While in the program editing or entry mode, simply press ZOOM, use the arrow keys to highlight the desired command, and then press ENTER. The command will be inserted into your program. **6:ZStandard** is particularly useful. It sets **Xmin** and **Ymin** to **–10** and sets **Xmax** and **Ymax** to **10**. The command **5:ZSquare** is useful when dealing with squares or circles that must appear without distortion. This command widens the current viewing range in one direction to make each pixel represent the same increment in both the x and y directions.

The ZOOM commands are less useful in setting up the window shown above, which has a range in each direction of **–3** to **8**. In this case, you must store the desired values to the window variables, as shown next.

Instruction	Explanation
:–3→Xmin	Be sure to use the (–) key, and not the subtraction key, for the negative sign. The arrow is entered with the STO> key. To enter **Xmin**, press VARS, press ENTER to select **1:Window** and then press 1 to select **Xmin**.
:8→Xmax	**Xmax, Ymin,** and **Ymax** are selected similarly to **Xmin**.
:–3→Ymin	
:8→Ymax	



Suggested Setup Commands

By now, you may have experienced the frustration of running the perfect program, only to find that the viewing window is set incorrectly to see the results, or that someone in another class period was using 2ND [STAT PLOT], resulting in an error statement or extraneous points on the screen. This Calculator Note suggests a few commands that can be used to avoid these difficulties.

Begin a new program called **SETUP**. Include in that program the commands from this list that you think may be useful.

PROGRAM:SETUP

Instruction **Explanation**

Window Option 1

:ZStandard This is the easiest way to set a viewing window with a range that extends from −**10** to **10** in both the horizontal and vertical directions. This can be useful when you are typing your setup commands directly into your main program. However, because these dimensions frequently must be modified for a particular program, in a separate setup program it is better to use the commands listed here, which set each window variable individually. To enter **ZStandard**, press ZOOM 6.

Window Option 2

:−010→Xmin Sets **Xmin** to **−10**. The leading zero makes it easier to edit this line for three-digit numbers after it is imported into your program. Be sure to use the negative key (−), and not the subtraction key. The arrow is entered with the STO> key. Enter **Xmin** and the other window variables in the next few lines by pressing VARS and then 1 to select **1:Window**, followed by the appropriate number key for the desired variable.

:010→Xmax

:-010→Ymin

:010→Ymax

Window Option 3

:ZSquare Adjusts the width of the current viewing window range in either the vertical or horizontal direction as necessary to prevent distortion to circles and squares. Because this command adjusts the Window variables, it should be used only after entering all other commands that affect those variables. Enter this command by pressing ZOOM and then 5.

Other Setting Options

:ClrDraw Clears the graphics screen. The ZOOM commands all clear the screen as well, so this may be unnecessary if a ZOOM command is used. Select **ClrDraw** from the 2ND [DRAW] menu.

:AxesOff (or AxesOn) Turns off (or on) the display of the coordinate axes. Press 2ND [FORMAT], move the cursor down to **AxesOff** (or **AxesOn**), and press ENTER.

:FnOff Turns off the display of the graph of any functions entered at the Y= screen. Press VARS, use the right arrow to highlight **Y-VARS**, press 4 to select **4:On/Off**, and then press 2 to select **2:FnOff**.

:PlotsOff Turns off the plotting of points entered in the STAT lists. Press 2ND [STAT PLOT] and then 4.

:Degree Sets the calculator to treat angle measurements as degrees rather than radians. Press MODE, use the arrow keys to highlight **Degree**, and press ENTER.

Using a Subroutine

A set of program commands that may be used by several different programs is often called a subroutine. This can be entered most efficiently as a separate program. The main program can then include a command to run the separate program as a subprogram, or the subroutine can be copied into each program that will use it.

The setup commands are a good example of a useful subroutine. Don't worry if the commands in your standard setup program don't exactly match what is needed for your current program. You can still use your setup program and then modify the commands as needed.

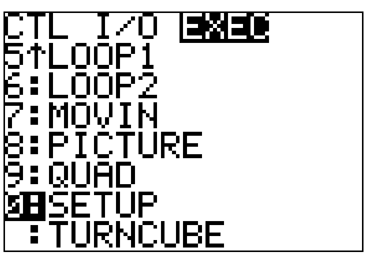
Calling a Subprogram

The easiest way to use your subroutine is to insert a command into your main program that tells it to run the subroutine as a separate program. The advantage of this method is that it is easy to use. The disadvantage is that you have to remember to keep the subprogram in your calculator; otherwise, your main program will not run correctly.

With the cursor at the first line of your main program (in the program entry or editing mode), press PRGM, use the right arrow to highlight **EXEC**, move the cursor down to the name of your setup program, and press ENTER. A command will be inserted into your program that tells it to run the setup program, as shown in the screen display here. If any of the setup program commands need to be modified, enter them after the command that calls the subprogram. For example, if your setup program turns the axes off, and you want them on, it is easier to enter these two lines into your program than to enter the entire list of setup commands individually.

:prgmSETUP

:AxesOn



Copying a Program

You can also use your setup program by importing a copy of it into your main program. This has the advantage of keeping everything you need for your program in the single program listing. This technique is useful as well for renaming programs or for copying a complex line repeatedly into a program.

While in the program entry or program editing mode, press 2ND [RCL]. Then press PRGM, use the right arrow to highlight **EXEC**, highlight the name of the program you wish to import, and press $\boxed{\text{ENTER}}$. A screen will appear similar to the one shown here. Press $\boxed{\text{ENTER}}$ again to copy all of the lines from the selected program into the main program. You can now go back and edit or delete individual setup commands as needed.

Storing and Recalling Pictures

Often, an animation program includes a stationary background that is displayed as the animated object moves in front of it. Because the animation requires that you repeatedly clear the screen, the need to constantly redraw the background can slow the animation down considerably, especially if circles are involved. These instructions use the *Arrow* activity as an example of how your program can store the background as a picture and instantly recall it after the screen has been cleared.

You can probably create this program most quickly by starting a new program, importing your own ARROW program into it, and then editing the program. See the instructions under *Copying a Program* in the Calculator Note "Using A Subroutine."

PROGRAM: ARROW2

Instruction **Explanation**

These setup commands are the same as for the original ARROW program, with the addition of **ZSquare**.

:ZStandard Quickly sets **Xmax** and **Ymax** to **10**. Press ZOOM 6.

:−3→Xmin **Xmin** is found on the VARS **Window** menu.

:−3→Ymin

:ZSquare Keeps the circles from being distorted. Press ZOOM 5.

:AxesOff This is found under 2ND [FORMAT]

:FnOff This is found under VARS **Y-VARS**.

:PlotsOff Press 2ND [STAT PLOT] 4.

Before beginning the animation loop, the background is drawn and stored as a picture.

:Circle(8,8,1) Draws a circle centered at (8, 8) with radius 1. Press 2ND [DRAW] 9 for the **Circle** command.

:Circle(8,8,2)

:Circle(8,8,3)

:StorePic Pic1 Stores the contents of the graphing screen as a picture, labeled **Pic1**. To enter **StorePic**, press 2ND [DRAW], use the right arrow to highlight **STO**, and press ENTER to select **1:StorePic**. To enter **Pic1**, press VARS, press 4 to select **4:Picture**, and press ENTER to select **1:Pic1**. You can take a shortcut here by using the numeral **1** in place of the variable, **Pic1**.

:For(A,1,8)

:ClrDraw

Because the screen is cleared as part of the animation, the background picture must be recalled each time the screen is erased.

:RecallPic Pic1 **RecallPic** is found immediately beneath **StorePic**. Press $\boxed{\text{2ND}}$ [DRAW], use the right arrow to highlight **STO**, and press 2 to select **2:RecallPic**. **Pic1** is again selected from the $\boxed{\text{VARS}}$ **Picture** menu.

:A–3→B

:A–1→C

:Line(B,B,A,A)

:Line(C,A,A,A)

:Line(A,C,A,A)

:For(D,1,30) Be careful not to use *A, B,* or *C* as the loop counter for this delay loop, because those variables are already in use.

:End

:End

Matrices

These instructions explain the use of matrices on the calculator, both from within a program and from without.

Manual Matrix Operations

First, we will look at how to create, edit, and use matrices without using a program.

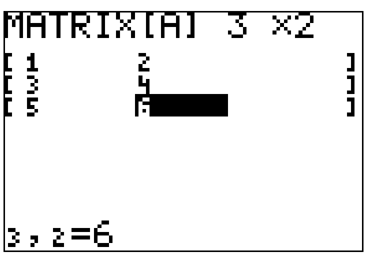
Creating a Matrix

As an example, assume that you wish to define the following matrix in your calculator:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

Press and use the right arrow to highlight **EDIT**. Press ENTER to select matrix **[A]** for editing. Because our matrix has three rows and two columns, we will define it as a **3 x 2** matrix. Press 3. (If another digit remains after the 3 because the matrix was previously defined with more than nine rows, use the DEL key to delete the extra digits.) Press ENTER. Enter the column dimension of **2** in the same way.

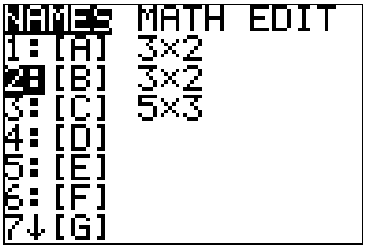
Enter the matrix elements row by row by pressing the number keys, followed by the ENTER key after each entry. After the last entry, press 2ND [QUIT] to exit the matrix editor.



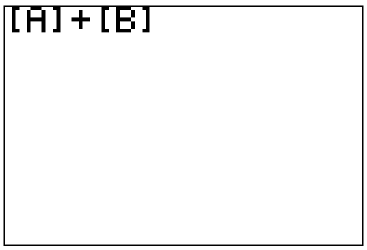
You can edit individual matrix elements by again selecting the matrix using 2ND [MATRIX] **EDIT** and then using the arrow keys to move the cursor to the element to be edited.

Matrix Operations

To select the name of a matrix for use in a mathematical operation, press 2ND [MATRIX], use the down arrow to highlight the name of the desired matrix in the **NAMES** menu and then press ENTER. The matrix name will be displayed, enclosed in brackets.



The display shown here would cause matrix **[A]** to be added to matrix **[B]**. The matrix name must be selected as described previously. If you instead enter it with the bracket symbols and the letter of the matrix name, the calculator will not recognize it as a matrix.



Matrices from Within a Program

Defining the Matrix

To define a matrix from within a program, the matrix elements must be listed row by row and then stored to the matrix name. Each row of the matrix must be enclosed in brackets, and the entire matrix must be enclosed in another set of brackets. (The brackets are found above the multiplication and subtraction keys using the 2ND key.) Matrix **[A]** from our previous example would be defined like this:

:[[1,2][3,4][5,6]]→[A]

Notice that elements within the same row are separated by commas, but there is no comma between rows. The → character is entered using the STO> key. Once again, you cannot enter the matrix name using the bracket symbols, but must select from the 2ND [MATRIX] **NAMES** menu.

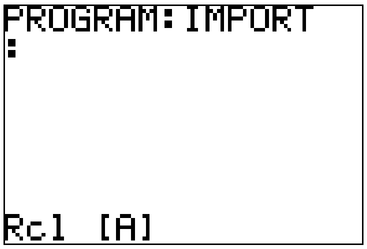
Importing the Matrix

If the matrix is lengthy, you might find it easier to create the matrix with the matrix editor, as described previously, and then import it into your program. To do this, press 2ND [MATRIX], use the right arrow key to highlight **EDIT**, select the matrix name, and press ENTER. Enter the matrix dimensions and the value for each entry. Exit the matrix editor by pressing 2ND [QUIT].

If you were within your program when you selected the matrix editor, exiting the matrix editor will also have exited you from your program. Press PRGM, use the right arrow to highlight **EDIT**, use the down arrow to highlight your program name, and press ENTER. Move the cursor to the position in your program listing where you wish to insert the matrix definition.

Import the matrix and assign it to a matrix name as described here.

1. Press 2ND [RCL]. ([RCL] is located above the STO> key.) **RCL** (recall) will appear at the bottom of the calculator screen.
2. Press 2ND [MATRIX], use the down arrow to highlight the name of the matrix under the **NAMES** menu, and press ENTER. Your screen will be similar to that shown here.



3. Press ENTER to import the matrix into your program.
4. Press STO> to direct the program to store the matrix.

5. Select the matrix name under which the matrix is to be stored by pressing 2ND [MATRIX] and once again selecting the name under the **NAMES** menu.

Addressing Individual Matrix Entries

In a program, it is often necessary to select an individual entry from a matrix. For example, our matrix **[A]** might represent a set of three ordered pairs describing the *x*- and *y*-coordinates of three points. If we wish to draw a line segment from the first point (row 1) to the second point (row 2), we must tell the program that the *x*-coordinate from which we are drawing is found in row 1, column 1 of the matrix and that the *y*-coordinate is found in row 1, column 2 of the same matrix.

The programming code for making reference to the entry in row 1, column 2 of matrix **[A]** is **[A](1,2)**. A command to draw a line segment from the first point in our matrix to the second point will look like this:

:Line([A](1,1),[A](1,2),[A](2,1),[A](2,2))

The four matrix entries in this command define, in order, the *x*- and *y*-coordinates of the first point followed by the *x*- and *y*-coordinates of the second point.

Defining Matrix Dimensions

Occasionally, you might need to store a value in an individual matrix element when the matrix itself has not yet been defined within the program. In that case, you must first define the dimensions of that new matrix. This is done with the **dim** command.

Note: When a matrix is created by listing the values for each element and then storing them to a matrix name, as described earlier, the matrix is dimensioned automatically, and it is not necessary to use the dim command. An example of this would be the program command **[[1,2][3,4]]→[A]**. However, if you wish to store a value to an individual entry of a matrix, such as **7→[B](2,1)**, and you have not yet defined matrix **[B]**, then you must use the **dim** command.

The syntax for defining matrix **[B]** as an 8-by-2 matrix is this:

{8,2}→dim([B])

Notice that the dimensions are enclosed in braces. These are the $\boxed{\text{2ND}}$ characters above the parenthesis keys. To select **dim**, press $\boxed{\text{2ND}}$ [MATRIX], use the right arrow to highlight **MATH**, and press $\boxed{3}$ to select **3:dim(**.

The TURNCUBE *Program*

When this program (shown on the next page) is run, the screen will show the cube turning (in 36 increments of 10° each). Here are the meanings of the key variables:

- Matrix A gives the vertices of the cube.
- Matrix B gives the coordinates of the projected vertices.
- Matrix C is the rotation matrix.
- The variables J, K, and L give the coordinates of the viewpoint.
- The variable S gives the location of the screen (as the plane z = S).
- The variables R and Q represent the ratio $\dfrac{z\text{-distance to the screen}}{z\text{-distance to the viewpoint}}$ for, respectively, the vertices on the back face of the cube (whose z-coordinate is 0) and the vertices on the front face of the cube (whose z-coordinate is 2).

The situation used in this program is specified in early lines of the program, as follows:

- The vertices of the cube are initially at (1, 0, 0), (1, 2, 0), (3, 0, 0), (3, 2, 0), (1, 0, 2), (1, 2, 2), (3, 0, 2), and (3, 2, 2).
- The viewpoint is at (5, 4, 10).
- The screen is the plane z = 4.

```
PROGRAM: TURNCUBE
:−3.7→Xmin
:5.7→Xmax
:−2.1→Ymin
:4.1→Ymax
:FnOff
:Degree
:[[1,0,0][1,2,0][3,0,0][3,2,0][1,0,2][1,2,2][3,0,2][3,2,2]]→[A]
:[[cos 10,sin 10,0][−sin 10,cos 10,0][0,0,1]]→[C]
:5→J:4→K:10→L
:4→S
:S/L→R:(S−2)/(L−2)→Q
:{8,2}→dim([B])
:For(C,1,37)
        :For(V,1,4)
                :[A](V,1)+R*(J−[A](V,1))→[B](V,1)
                :[A](V,2)+R*(K−[A](V,2))→[B](V,2)
        :End
        :For(V,5,8)
                :[A](V,1)+Q*(J−[A](V,1))→[B](V,1)
                :[A](V,2)+Q*(K−[A](V,2))→[B](V,2)
        :End
        :ClrDraw
        :Line([B](1,1),[B](1,2),[B](2,1),[B](2,2))
        :Line([B](2,1),[B](2,2),[B](4,1),[B](4,2))
        :Line([B](3,1),[B](3,2),[B](4,1),[B](4,2))
        :Line([B](1,1),[B](1,2),[B](3,1),[B](3,2))
        :Line([B](5,1),[B](5,2),[B](6,1),[B](6,2))
        :Line([B](6,1),[B](6,2),[B](8,1),[B](8,2))
        :Line([B](7,1),[B](7,2),[B](8,1),[B](8,2))
        :Line([B](5,1),[B](5,2),[B](7,1),[B](7,2))
        :Line([B](1,1),[B](1,2),[B](5,1),[B](5,2))
        :Line([B](2,1),[B](2,2),[B](6,1),[B](6,2))
        :Line([B](3,1),[B](3,2),[B](7,1),[B](7,2))
        :Line([B](4,1),[B](4,2),[B](8,1),[B](8,2))
        :[A]*[C]→[A]
:End
```