Activity 1B: Programming Animations I

Students enter a program into their calculators in order to gain first-hand experience with a process similar to computer animation. This activity exposes students to the connection between computer programming, animation, and mathematics.

> **Note:** In Part 4, as an optional activity, students work with a calculator program that uses matrix transformations. At that point, students should be able to understand the logic of the program and see how the transformations they learned about are applied in the calculator program.

Sequence

1B.1: ***Programming the Calculators***	Students transfer code into their graphing calculators to create a simple animation.
1B.2: ***Class Discussion***	The class discusses the functions of the commands in the calculator program.

Understandings

- Calculators can be programmed to create a simple animation.
- Programming a calculator is similar to some processes used in computer programming.

Materials Needed

- **Handout 2: TI Calculator Animation Program #1**
- T1 graphing calculators and manuals (1 per student)
- Black link cables (1 per student)

> **Note:** The cables should have come with the graphing calculators when they were purchased.

1B.1: Programming the Calculators

1. Distribute materials.

Give each student **Handout 2: TI Calculator Animation Program #1**, a graphing calculator and manual, and a black link cable.

Teacher's Notes: TI Calculator Animation Program #1

This program uses a loop and a pair of line segments to create the illusion of a moving smile. The program uses a translation to slide the picture from the upper left corner of the calculator screen to the lower right corner, all within the first quadrant.

2. Have students transfer the program into their calculators.

Students can use the instructions provided in the handout or they can download the program from your calculator (or a classmate's calculator) using the link cables.

3. Have students run the program.

After running the program, encourage students to discuss what they see on their calculators with another student.

Note: For more information on programming calculators and transformations, see *Additional Resources for Teachers*. As an extension to Handout 2, have students edit the program to make the 'smile' move in a different direction, or more smoothly, or over a longer period of time.

Handout 2:
TI Calculator Animation Program #1

You can experiment with the programming capabilities of a TI graphing calculator to create short animations.

Enter Program #1 below into your graphing calculator, run the program, and discuss what you see with your classmates.

You can also experiment with other DRAW functions to create different images. You may want to consult the calculator manual to determine the parameters for each new function.

Program #1: Moving Smile

To create a new program:

- Press PRGM on the keypad of the graphing calculator.

Move the cursor to NEW and press ENTER.

Enter a name for the first program (such as "MOVETRIANGLE") and press ENTER.

> **Note:** The graphing calculator is automatically in ALPHA mode in order to allow you to type letters. The letters are written in blue or green above individual calculator keys.

Wait for the screen to display a colon, then enter each command below. Press ENTER after each new command.

Command	Function and Location
:0 → Xmin	Sets up the viewing window. The triangle in this program will only move within the first quadrant of the coordinate system. The arrow assigns the number 0 to **Xmin** by using the store key, **STO>**. This value will stay in **Xmin** until reassigned. *To find this command:* Use the Window menu within the **VARS** key.
:25 → Xmax	
:0 → Ymin	
:25 → Ymax	
:FnOff	Turns off the display of any equations entered in the **Y=** editor. *To find this command:* Press **VARS**, move the cursor to **Y-VARS #4**, and press **ENTER**.
:PlotsOff	Turns off all plots. *To find this command:* Press **2nd** **[STAT PLOT]** **4**.

Command	Function and Location
:AxesOn	Makes the coordinate axes visible during the animation. *To find this command:* Press 2nd [FORMAT].
:1 → P **:20 → Q** **:4 → R** **:16 → S** **:4 → T** **:16 → U** **:7 → V** **:20 → W**	Initial values are stored within variables: • *P* and *Q* represent the *x*- and *y*-coordinates of one endpoint of the segment. • *R* and *S* represent the *x*- and *y*-coordinates of the second endpoint of the first line segment. • *T* and *U* represent the *x*- and *y*-coordinates of one endpoint of the second segment. • *V* and *W* represent the *x*- and *y*-coordinates of the second endpoint of the second segment. Use the Alpha keys to enter letters. Use the **STO>** key to assign values to each.
:For(N, 1, 8) **:ClrDraw** **:Line(P, Q, R, S)** **:Line(T, U, V, W)** **:For(A, 1, 50)** **:End** **:P+3 → P** **:R+3 → R** **:T+3 → T** **:V+3 → V** **:Q-2 → Q** **:S-2 → S** **:U-2 → U** **:W-2 → W** **:End**	This begins a loop with counter *N* that ranges from 1 to 8. The loop repeats eight times. *To find* **For** *and* **End**: Both are listed in alphabetical order in the calculator's CATALOG. *To find* **ClrDraw** *and* **Line**: Press **2nd [DRAW]**, **#1**, and **#2**. The **Line** function sketches the two segments that form the object in this program; the input of this function includes the *x*- and *y*-coordinates of each segment's endpoints. **ClrDraw** at the beginning of the **For** loop clears the previous image from the screen. Each coordinate of the two endpoints of the line segments is changed: The segments are *translated* (shifted) to the right three units and down two units a total of eight times, completing the **For** loop.

To run the program:

- Press PRGM.

Move the cursor over the name of the program.

Press ENTER. The program is executed.

1B.2: Class Discussion

1. Discuss the function of each program command.
Walk students through the functions used in different lines of the code so that students can understand how the program creates a simple calculator animation. Ask:

- What command in the program shifts the two line segments?
- What does the For loop do? How might you write this program without a For loop?

Teacher's Notes: Understanding the Calculator Programming

It is not necessary to be an expert at programming with TI calculators to lead or take part in this discussion. Having students look closely at the program allows them to see its different elements and gives them a feel for how a short animation can be created. This is especially true if students enter the program commands directly into the calculator themselves.

If some students in the class have had experience with programming, they may be a good resource for the rest of the class. For example, a student might explain what a loop does and how it functions within the program.

2. Have students consider flipbook animation.
Tell students that they will create a flipbook animation as part of their unit project. Ask the following questions to help students make connections between the program's commands and flipbook animations:

- How could the program be changed so that the figure appears to move across the screen more rapidly?
- What have you learned that might help you when you create a flipbook for your unit project?