INTRODUCTION TO COMPUTER ANIMATION WITH YOUR TI-83 PLUS

Kenneth C. Wolff
Montclair State University
Department of Mathematical Sciences
Upper Montclair, NJ 07043
wolffk@mail.montclair.edu

In the past decade the use of powerful computers has revolutionized the making of animated moves and numerous television advertisements. Films such as Toy Story, Toy Story 2, A Bug's Life, Shrek and Monsters, Inc. have made the public aware of the high quality and sophistication that computer graphics, powered by modern high-speed computers, can deliver. However, most of the viewing public, which includes our students, is not aware of the mathematics that drives computer graphics. The mathematics that supports computer graphics and animation is a natural connection between the mathematics classroom and the world of entertainment. This article presents the basic concepts and tools that are used to implement elementary computer graphics on a TI-83 Plus graphing calculator. The material can be used with students who have attained different levels of mathematical maturity. Explanation of material can be at a very intuitive level or based upon more abstract mathematical models. Although this article uses terminology specific to Texas Instrument (TI) graphing calculators, the material can be adapted for use with other graphing calculators such as those available from Casio, Hewlett-Packard or Sharp.

The author assumes the reader is familiar with some features of the TI-83 Plus calculator but not with the use of the features of the DRAW, MATRIX, and PRGM menus. The material is presented as a guided discovery lesson and has been found to work best when students work in pairs. Approaches to topics and program development are presented in a manner consistent with the author's classroom and workshop experience.

Basic Features of the DRAW, MATRIX and PRGM Menus

Before we start to explore the drawing of simple figures it is suggested that we take care of some general calculator housekeeping by going to the MODE menu and selecting all of the leftmost options. Next set the viewing window to [-2,10] by [-2,10], select AxesOff from the FORMAT menu and, if necessary, turn off the graphs of any active functions or statistical plots.

We begin our explorations by using the DRAW features of the calculator to display the "stick figure" shown in Figure 1 on the next page. The sequence of four commands that produce the figure is shown in Figure 2. As shown in Figure 2, from the home screen go to the DRAW menu and select the ClrDraw command. Paste it to the home screen by pressing the ENTER key. Press the ENTER key again to execute the command. Return to the DRAW menu, select the command Line(, paste it to the home screen and edit it so it

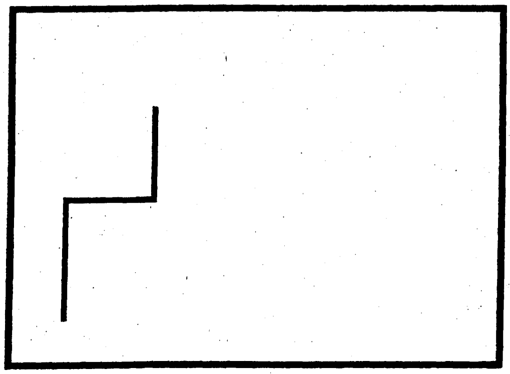
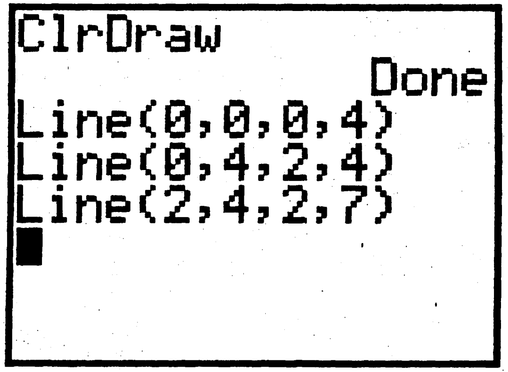


Figure 1
Vertices at (0,0), (0,4), (2,4) and (2,7)



```
ClrDraw
                 Done
Line(0,0,0,4)
Line(0,4,2,4)
Line(2,4,2,7)
█
```

Figure 2
Commands to create Figure 1

reads Line(0,0,0,4). Pressing the ENTER key executes the command and should result in a vertical line segment being drawn on your calculator's screen. Return to the home screen and repeat this process so that you have edited and executed the remaining two Line(commands. The end result should look like Figure 1.

To assess student's understanding of the use of these commands ask them to draw the triangle with vertices at (2,0), (7,0) and (5,6) as shown in Figure 3. Follow this with a discussion about how they would draw a more complicated figure such as the house shown in Figure 4 or how they might create multiple copies of a figure at different locations on the screen.

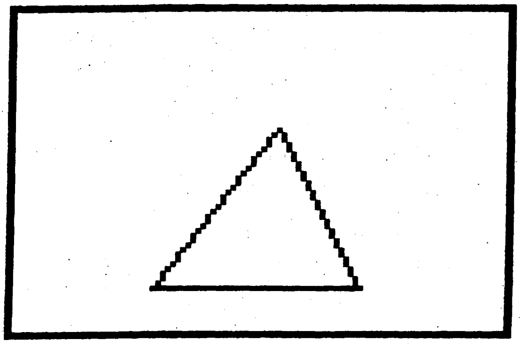


Figure 3
Vertices at (2,0), (7,0) and (5,6)

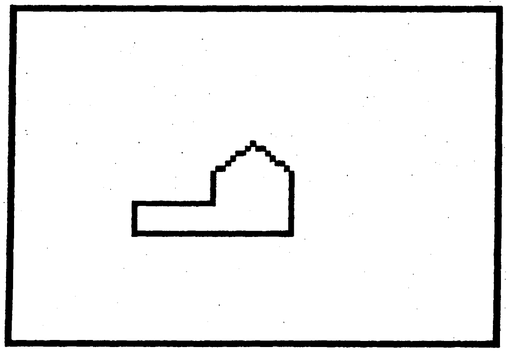
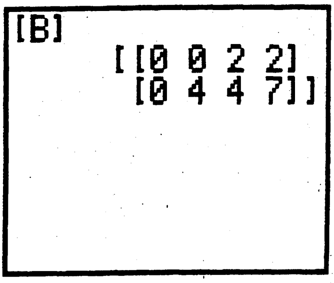


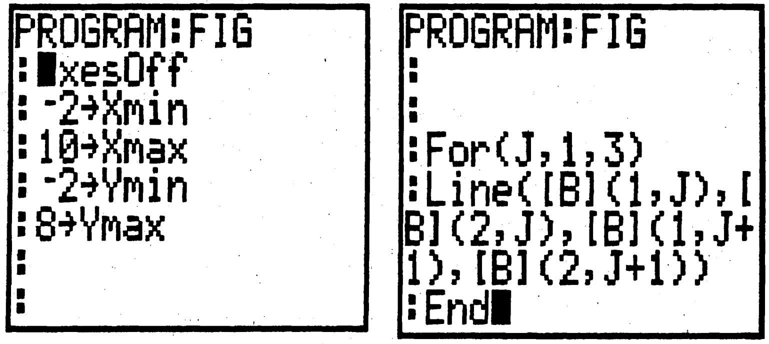
Figure 4
Vertices at (2,1), (5,1), (5,3), (4,4), (3,3), (3,2), (1,2) and (1,1)

Hopefully you will be pleasantly surprised by the suggestions of students such as the need to repeat the use of the Line(command many times and a convenient way to keep track of a figure's coordinates. Thus, at this point, it is natural to introduce the use of a matrix to keep track of the coordinates of a figure and a program with a loop to draw the line segments that make up the figure. The matrix [B] shown in Figure 5 and the program FIG shown in Figure 6 can be used to draw the stick figure shown in Figure 1.



```
[B]
      [[0 0 2 2]
       [0 4 4 7]]
```

Figure 5
Matrix [B]



```
PROGRAM:FIG
:█xesOff
:-2→Xmin
:10→Xmax
:-2→Ymin
:8→Ymax
:
:
```

```
PROGRAM:FIG
:
:
:For(J,1,3)
:Line([B](1,J),[
B](2,J),[B](1,J+
1),[B](2,J+1))
:End█
```

Figure 6
Two screen dumps that show Program FIG

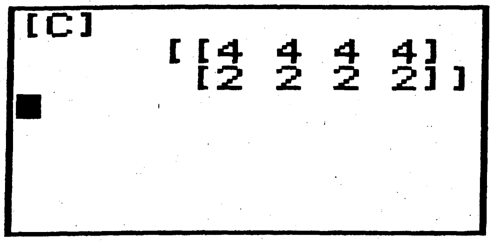
Note that [B](1,J) denotes the element in the first row and Jth column of matrix [B]. At this time it is probably necessary to spend some time explaining how to enter and edit values in a matrix and how to enter, execute and edit a program such as FIG, which is described below. Although the author recommends that students be given the experience of entering the program commands, an alternate approach is to write the program on one calculator and then distribute it to students' calculators using their linking capabilities. Whichever approach is used, it is important to spend time analyzing the steps in the program.

To create the program FIG, first go to the PRGM menu and select NEW. Type the program name and touch the ENTER key. Most commands such as ClrDraw and AxesOff are obtained by going to appropriate menus such as DRAW and FORMAT, selecting the desired commands and by pressing the ENTER key which pastes the command into the program. The -> symbol is entered by pressing the STO key. Window variables are found by first going to the VARS menu and then the Window sub-menu. The For(command is found under the CTL sub-menu of the PRGM menu. Names of matrices such as [B] must be entered using the NAMES sub-menu of the MATRIX menu and not by using the [and] brackets from the keypad. Readers should consult their calculator manual for more details as necessary. Once students have successfully used the program FIG to reproduce Figure 1 their understanding can be assessed by having them modify matrix [B] to draw the triangle shown in Figure 3 or the house shown in Figure 4. For either of these figures students quickly discover the need to repeat the coordinates of the initial vertex in the last column of the matrix. The may also decide to change the graphing window by editing program FIG.

As outlined in the remainder of this article, program FIG in combination with other short programs and appropriately defined matrices can be used to transform graphical figures while at the same time maintaining students' interest and reinforcing important mathematical concepts. Using the stick figure shown in Figure 1 to introduce a new type of movement before trying it with more complicated figures provides some continuity and basic references for new learners.

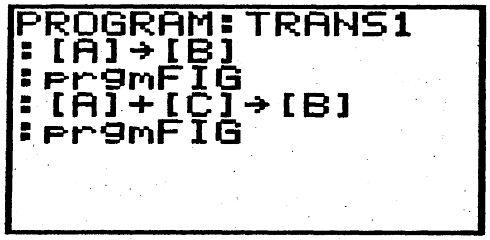
Translations

Given some appropriate guidance students can usually explain the relationship between the coordinates of a plane figure such as a triangle and a translation of that figure. For example, consider the triangle in Figure 3. Suppose we want to draw that triangle and a copy of it translated 4 units in the positive x-direction and 2 units in the positive y-direction. The new figure will have vertices (2 + 4, 0 + 2) = (6, 2), (7 + 4, 0 + 2) = (11, 2) and (5 + 4, 6 + 2) = (9,8). Thus to draw the original triangle and the translated copy we could run the program FIG, edit the matrix [B] and then rerun the program. Students quickly see the obvious drawbacks to this approach if we want to drawn many translated copies of the triangle. Their experience with the loop in program FIG can be used to motivate the introduction of a translation matrix [C] as shown in Figure 7 and the program TRANS1 shown in Figure 8.



```
[C]
        [[4  4  4  4]
         [2  2  2  2]]
▮
```

Figure 7



```
PROGRAM:TRANS1
:[A]→[B]
:prgmFIG
:[A]+[C]→[B]
:prgmFIG
```

Figure 8

Notice that we use matrix [A] to store the coordinates of the original figure, program FIG and matrix [B] to draw the figures, and matrix [C] which is added to matrix [A] to create a new matrix [B] with the coordinates of the translated figure. The results of executing the program are shown in Figure 9. After running TRANS1 students may want to change the graphing window by editing program FIG. They may also want to discuss and experiment with the use of the ClrDraw command. Should it be judiciously placed in a program or executed as needed from the home screen? At this time, students need to experiment with what they have learned by translating other figures. A rich assessment activity is to have them modify the programs FIG and TRANS1 so that their calculators draw the original triangle shown in Figure 3 and five translated copies. One way of accomplishing this is to use the edited program FIG1 as shown in Figure 10 and the program TRANS2 shown in Figure 11. The result is shown in Figure 12.

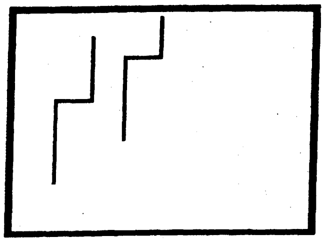
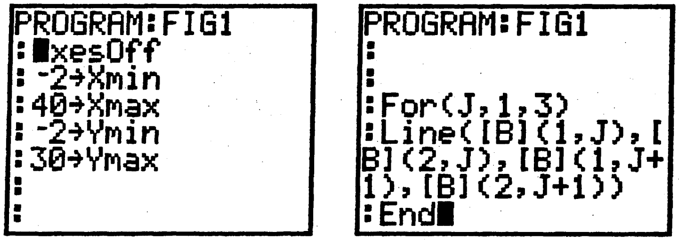


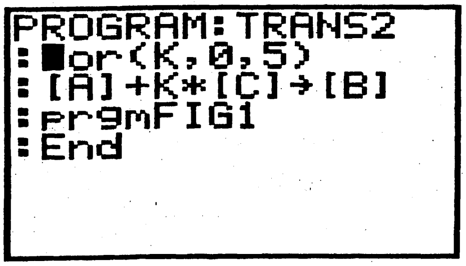
Figure 9
TRANS1 applied to the
basic stick figure



```
PROGRAM:FIG1
:▮xesOff
:-2→Xmin
:40→Xmax
:-2→Ymin
:30→Ymax
:
:
```

```
PROGRAM:FIG1
:
:
:For(J,1,3)
:Line([B](1,J),[
B](2,J),[B](1,J+
1),[B](2,J+1))
:End▮
```

Figure 10
Program FIG modified for use with TRANS1
and TRANS2



```
PROGRAM:TRANS2
:█or(K,0,5)
:[A]+K*[C]→[B]
:prgmFIG1
:End
```

Figure 11
Program TRANS2

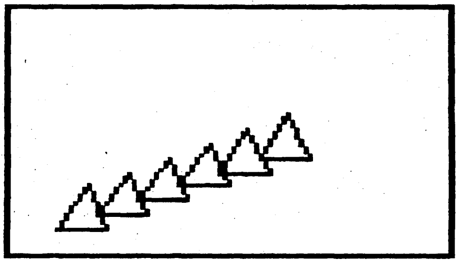


Figure 12
The result of program TRNS2

Creating Motion

Returning to the previous comments about the use of the ClrDraw command, students may want to create a sense of motion by using it to erase a figure before the transformed figure is drawn. If the transformed figure is being drawn too fast a delay loop be introduced. For example experiment with the effect of inserting the lines :For(N,0,300) and :End between the third and fourth lines of TRANS2. Experiment with the amount of delay by changing the value of 300 to 700.

Other Transformations

When we change the size of a figure by tripling or halving all of the dimensions we have *scaled* our figure by a factor of 3 or 1/2, respectively. When a fence post is pushed out of a vertical position, a point at the top of the post moves a greater horizontal distance than a point near the bottom of the post. This type of transformation is called a *shear*. Both of these transformations can be accomplished by multiplying the coordinate matrix for a figure by an appropriate square matrix. The same is true for *projections*, *reflections* and *rotations*. An exploration of the use of homogeneous coordinates leads to the discovery that all of these transformations, including translations, can be implemented with matrix multiplication. There are many extensions to these ideas, including implementing them on a computer where the quality of the graphical output is superior to what is possible with a calculator.

References

Crowley, Mary L. "Transformations: Making Connections in High School Mathematics," 79 – 91. 1995 Yearbook, Connecting Mathematics across the Curriculum, edited by Peggy A. House and Arthur F. Coxford, NCTM.

Herman, Eugene A., Michael D. Pepe, Robert T. Moore, and James R. King. Linear Algebra Modules for Interactive Learning Using MAPLE. Preliminary Version. Addison-Wesley, 2000.

Lay, David C. Linear Algebra and Its Applications. Second Ed. Addison-Wesley, 1997.

Wolff, Kenneth C. "Elementary Graphics and Animation with Your Calculator." Mathematics Teacher 95, (March 2002):172-176. NCTM.