Transformation Game

Teacher Notes and Answers

7 8 9 **10 11 12**

 TI-Nspire CAS

 Investigation

 Teacher

 40 min

Introduction

This activity, designed to be conducted as a 'whole class' activity, requires the teacher to program TI-Nspire CAS to generate a random graph from a predefined family of curves. Students are then asked to determine the equation of that randomly generated graph. It requires an IWB or data projector so that students may easily view the calculator screen (i.e. via TI-Nspire CAS software). There is a student worksheet in which students can record their answers.

Making the game file

The template is designed to support a curve guessing game for students Years 10-12 who may be studying polynomial functions, circular functions, exponential or logarithmic functions (or others). It makes use of only five commands. The first four commands initialise the game, and the fifth command generates and stores a random curve of the predefined type.

As an example, we will look at some transformations of the basic sine function.
Specifically, we will use three transformation parameters as follows:

$$f(x) = a\sin(bx) + c$$

To create the game template file using the TI-Nspire CAS, follow these steps.

- Press (home) **> New Document**, and then select **Add Calculator**.
- Type the command **a:={−3,−2,−1,1,2,3}** and then press enter
- Type the command **b:={1/2,1/4,1,2,3,4}** and then press enter
- Type the command **c:={−3,−2,−1,0,1,2,3}** and then press enter

(Note: These commands set the possible values for the transformation parameters a, b, and c in the function. The syntax ":=" means "assign to", and allows values to be stored in a variable. It can be entered by pressing ctrl *).*

- Type the command **curve:={}**

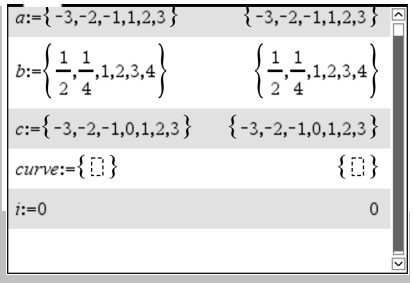
(This command creates and clears the contents of a list variable called 'curve', which will be used to store a list of the sine function rules used in each game.)

- Type the command **i:=0**

(This command creates and assigns the value '0' to a variable called 'i', which will be used to store the count number of each rule (i.e. the current question number in the game).

Author: D. Tynan

Tip: If you are playing the game repeatedly, it is convenient to combine all four of these commands into a single entry (separate each command with a colon character ":"). This means that all the initialisation, including making any changes to the permitted set of values for *a*, *b*, *c* can be made within a single entry.
The relevant command would be typed as follows:

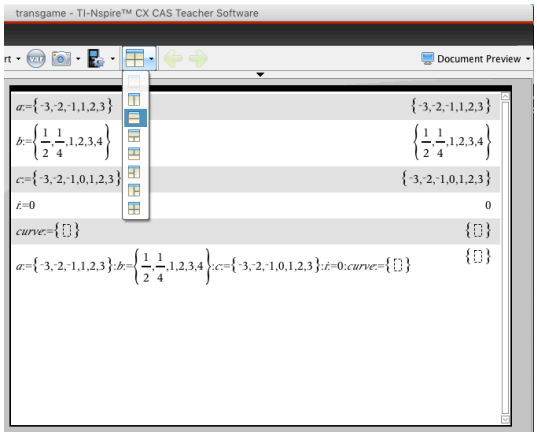


a:={−3,−2,−1,1,2,3}:b:={1/2,1/4,1,2,3,4}:c:={−3,−2,−1,0,1,2,3}:i:=0:curve:={}

The next step is to split the window so that you can generate a new graph with each press of the enter key (which works recursively).

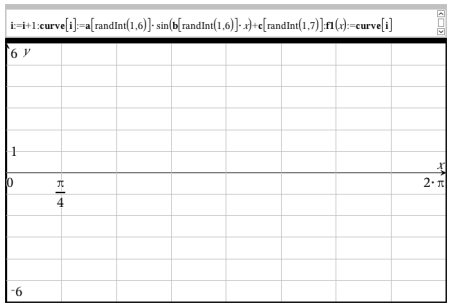
If you are using the *TI-Nspire CAS Teacher software* in Document Preview, select the split window option as shown.

If you are using the handheld, press doc ▾ **> Page Layout > Select Layout > Layout 3** to select the split window as shown.

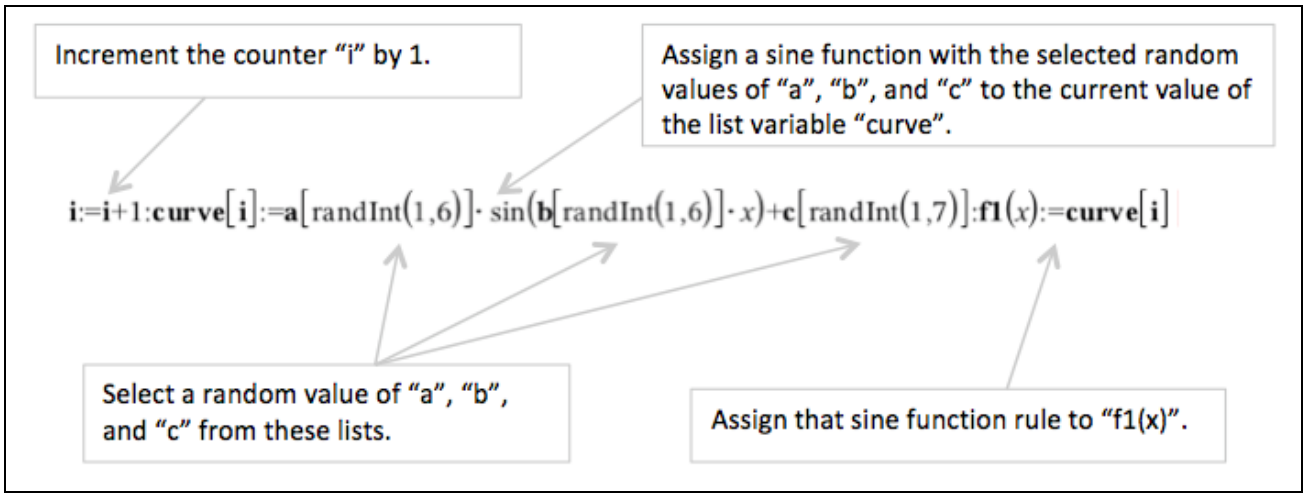


To set up the graph part of the screen

- Press ctrl tab to switch to the bottom half of the new 'split' window
- Select **Add Graphs**.
- Press menu **> Window/Zoom > Window Settings** to make the graph screen have dimensions [0,2π] by [–6,6], and set Xscale = π /4 and Yscale = 1
- Press menu **> View > Grid > Lined Grid** to make a lined grid visible (see result right).



The final command is to generate the 'random' sine curve (random within the defined values of *a* and *b*). This command is explained in the screen below.

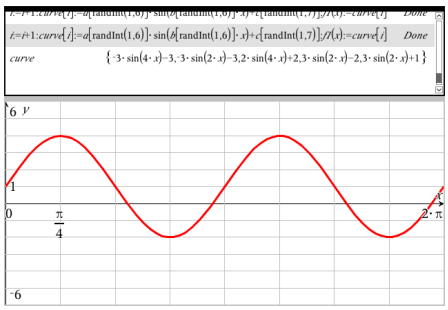


Author: D. Tynan

Now click in the top of the split window (use ctrl tab), and type this command:

i:=i+1:curve[i]:=a[randInt(1,6)]*sin(b[randInt(1,6)]*x)+c[randInt(1,7)]:f1(x):=curve[i]

Press enter to plot the first line. If no plot appears, press ctrl tab to select the graph part of the split window, and then press ctrl G. Locate the rule for **f1(x)** and press enter.
The graph will then be displayed as shown right.



To get a new graph, press ctrl tab to move to the **Calculator** page, and press enter. With each press of the enter key, a new 'random' function is created and its graph is drawn.

[Note: The position of the horizontal 'split' line can be adjusted by hovering over the line and clicking it and dragging it to the desired position.]

When you have finished a group of lines (five lines works pretty well), and you wish to check answers, type "**curve"** to display a list of the curves generated (the variable 'curve' displays the right hand side of equation *y* = a sin(bx)+c).

$$curve \qquad \{-3\cdot\sin(4\cdot x)-3,-3\cdot\sin(2\cdot x)-3,2\cdot\sin(4\cdot x)+2,3\cdot\sin(2\cdot x)-2,3\cdot\sin(2\cdot x)+1\}$$

The screen shows the five expressions used from a 'five-curve' game.

Note: To reset the game, remember to first copy and paste the command, before entering the curve 'generator' command. This restates the saved parameters, and initialises the two key variables.
a:={−3,−2,−1,1,2,3}:b:={1/2,1/4,1,2,3,4}:c:={−3,−2,−1,0,1,2,3}:i:=0:curve:={}

Author: D. Tynan

Teacher notes

- This task is intended to be attempted after students have completed at least some work on the function type, or have covered work on transformation such as dilations, reflections and translations.
- The focus is on the graphs of reinforcing students ability to quickly determine the rule for a function from the graph – and interpreting its shape and location as transformations of a basic graph for that function type.
- The finished file is included with this task, but the value of creating yourself as a teacher is that you will be clearer about how to modify the parameter values or window dimensions to suit your intended focus.
- It is intended that this be a whole class activity using the TI-Nspire CAS Teacher Software and an IWB. Once the game has been completed a few times, it is possible for students to pair up and challenge each other.
- From experience, it is worth repeating the game until most students get '5 out of 5'.
  As an extension, you can reduce the time permitted for each graph.
- A side benefit of this task is that it demonstrates to students how a limited set of commands can be combined from the calculator page to mimic common programming ideas (eg initialisation, looping, storing of the results of loop iterations)

Author: D. Tynan