Basic Programming for the CASIO 9850 Series of Calculators
Graphics

By Martin Schmude

This is the second installment to Basic Programming on the CASIO 9850 series of calculators and deals with graphics. Over the following lessons, you will use what you've learned in the previous booklet and be introduced to new features.

Unlike the previous section, the final product will be a collection of impressive and fun graphics. We will use one of these to include in our SUMDICE program that we wrote in the previous booklet. This will help interest the user and act as a nice starting point.

The rest of the work in this booklet will be up to you to use and manipulate where necessary. So feel free to experiment, these programs are yours. In the Appendix, you'll find the code to all the programs used in this booklet.

Any feedback or comments, as well as further questions would be very welcomed. I hope you enjoy the learning experience.

Marty Schmude

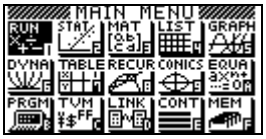
CONTENTS

SECTION 2

LESSON 1 – Setting the ViewWindow

The window on the CASIO *CFX9850GB PLUS* has 8001 little pixels (the word 'pixel' is an amalgamation of the words *picture element*). Each pixel can be turned on or off, and unlike other graphic calculators, can be a certain colour (green, blue and orange). Each pixel has a coordinate and in this lesson, we are going to adjust the window in such a way that it will be very simple to name those coordinates.

1. Enter RUN mode from the main menu. You can do this by moving the cursor to RUN and pressing **EXE,** or you can simply press 1, because of the 1 at the bottom of the RUN icon.

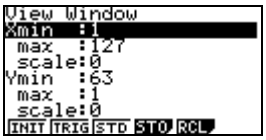


2. On your keypad, notice above the F3 button the word **V-Window**. This stands for ViewWindow and it is in here that you adjust the scale of your graphics window. Also note that V-Window is in yellow, which means you need to press **SHIFT** to enter the settings menu. Enter this menu now.

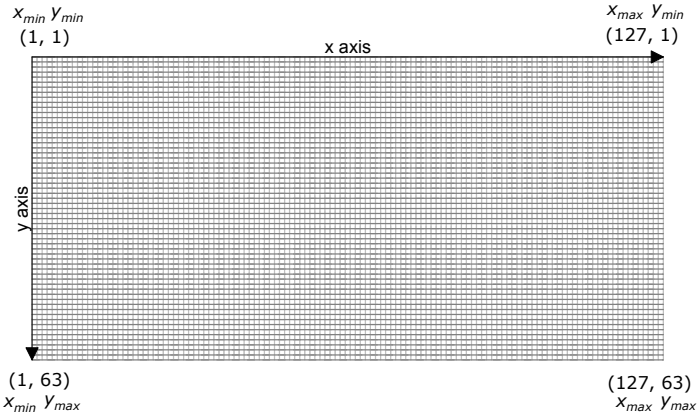
3. The menu is in two main parts, the *x*-axis and the *y*-axis.

   For your information, there are 127 pixels across (x-axis) and 63 pixels down (y-axis).

   This means the easiest scale to choose is one where the coordinates of each pixel will be an integer. Adjust the settings to mirror those shown below.



   The diagram below shows how the axes are scaled now. This will be the setting kept for this entire booklet. Notice how the origin is in the top left corner of the screen, and is commonly used in computers.



How it works

Let's figure out what each of those settings really mean. The ViewWindow is interested in three things for the *x*-axis.

1. The largest *x* value, located on the right-hand side ($Xmax$).
2. The smallest *x* value, located on the left-hand side ($Xmin$).
3. The scale on the axis, which means how often to place a tick on the line (scale). In our case, we set it to 0, as we do not wish to have any ticks.

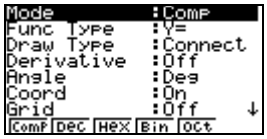
This works similarly with the *y*-axis.

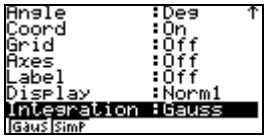
LESSON 2 – Adjusting peripheral settings

There are other settings that we will adjust but are not absolutely crucial. At worst, they get in the way and can detract from the picture.

1. Enter RUN mode from the main menu.

2. Each mode on the calculator has a 'brain', and it is there that we do the adjusting. Notice above the **MENU** key is the word **SETUP**. SETUP is the 'brain'. Press **SHIFT SETUP** and a list of settings is displayed.

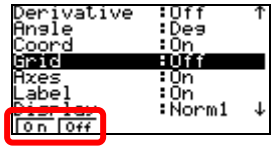




3. The settings we are interested in are Grid, Axes and Label, and we are going to turn them all OFF.

   - Grid – places dots on the screen at integer values on the number plane.
   - Axes – displays the *x* and *y*-axes.
   - Label – places the letters *x* and *y* at the end of the *x* and *y*-axes respectively.

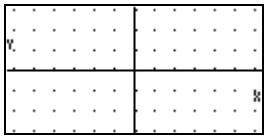
   To turn them off, move the cursor (using the up and down arrow keys) so that it is on Grid. Options appear at the bottom of the screen. Press OFF (F2).



4. Continue with the Axes and Label settings and turn them OFF.

How it works

Below is a picture of the three settings when they are turned on. You can see why we turn them off.



LESSON 3 – Working with Individual Pixels and the Sketch Menu

We are going to turn some pixels on and off. This will also help you become familiar with the coordinates of the screen. The syntax of the command we will be using is perhaps different to what you might expect. When we name the pixel we want to show, we give the *y*-coordinate first, then the *x*-coordinate.

PxlOn *y*, *x*

Unfortunately, other commands want the *x*-coordinate first. To make it simpler as we go through, subscripts will be used to help define the coordinate.

1. If you are not already in RUN mode, enter it now from the main menu.

2. The most important menu we will use throughout this booklet is the **SKETCH** menu located above F4. Enter this menu now, by pressing **SHIFT** first.

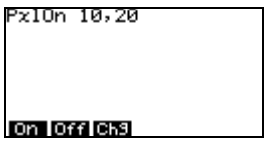
3. Press 'des' (F6) to cycle through the possible options.

4. Press 'des' (F6) again until you can see the option PIXL (F3).

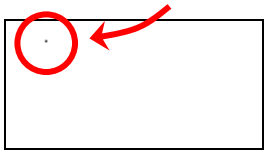
   Just for your information, notice how the bottom right corner of the PIXL option is cut off. This tells the user that this will open to another menu. When the option is a complete rectangle, like the TEXT option, this means it is an actual command to execute.

   Press PIXL (F3) and you will see three options: On (F1), Off (F2) and Chg (Change, F3). Press PxlOn (F1) followed by 10, 20.

   We are going to turn the pixel located at (10_y,20_x) on. The comma is located above the grey **DEL** button on the keypad (mid-right).



   Press the **EXE** button and notice the single pixel is on. Notice the coordinate is (10_y,20_x) i.e. 10 pixels down and 20 across.



5. Press the **AC/ON** button to exit the screen. Now do the same as above, but with the Off (F2) option. The result will be a blank screen, which is what is expected.

Why Not Try…

1. See if you can build a little picture using these simple commands. Perhaps sketch your initials, or a love heart, or PS2 emblem.
2. Experiment with the `Chg` (F3) option. This means change the pixel, so if it on, it will turn it off and vice versa.

LESSON 4 – Cls and the F-Line command

One the most used commands from now on will be that which joins two points together to form a line. The simplest command on the calculator to do this is called the F-Line command. The syntax of this command is as follows: imagine we wish to draw a straight line from $A(x_1,y_1)$ to $B(x_2,y_2)$, then the command line reads:

F-Line x_1, y_1, x_2, y_2

Obviously this is the same as joining B to A.

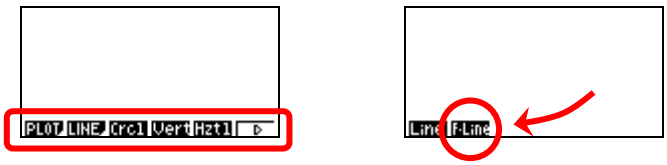
We are also going to introduce the method of instantly clearing the screen.

1. Enter RUN mode.

2. Ensure that the ViewWindow settings are the same as the previous lessons.

3. No doubt you have some work left over from the last **Why Not Try**. It is necessary then to clear the screen. There is an option that does this and it is found in the **SKETCH** menu. You will see the command Cls (F1), which means 'clear screen'. Press Cls (F1), followed by the **EXE** button. A 0 is displayed to acknowledge the clearing.

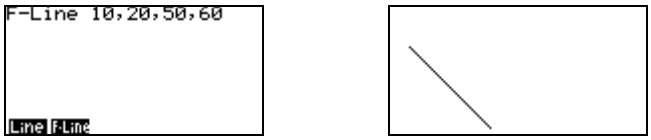
4. While you are in the SKETCH menu, locate the option LINE (F2). Press LINE and you will see another menu, which contains an option called F-Line (F2).



5. The syntax for this command is labeling the two points you want to join. For example, if we want the join the point (10,20) with the point (50,60), it would be entered like:

F-Line 10,20,50,60

Let's enter that now. You should end up with the following graphic.



Why Not Try…

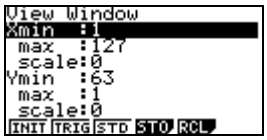
1. Once again, experiment with this command and see if you can draw something interesting. Use your imagination!

LESSON 5 – Entering Text

It is possible to draw many other things. They can be things related to graphs like tangents and inverse functions, or they can be horizontal and vertical lines. This lesson is going to teach you how to enter text in your diagrams. Remember to use the `Cls` command to clear your screen when you want.

1. Enter RUN mode.

2. Ensure that the ViewWindow settings are the same as shown below.



3. Press EXIT and open the **SKETCH** menu. Clear your screen by executing a `Cls` command (F1). Now press 'Des' twice (F6) until your screen looks like the one below.

4. One of the other useful commands you can use on the calculator is typing text, using the `Text` command (F2). Unlike the text we've used in the other booklet, this text is purely graphic text. That is, once it is entered, it cannot be erased[1]. To enter text, you must tell the calculator the coordinate where you want the text to start, and then what the text is. The syntax is:

```
Text y,x,"type text here"
```

Press Text (F2) and then type in the point $(10_y,20_x)$. Follow this with `"PROGRAMS ARE EASY"` so the line should look like below. Press **EXE**

Notice that the calculator place the top left of the letter `P` at to exact point $(10_y,20_x)$.

Why Not Try…

1. Clear your screen with `Cls` (F1). Experiment with the `text` command and write some text like your name.

[1] You could use `PxlOff` which would involve turning off every single pixel in the text.

LESSON 6 – Drawing Graphics with Programs: Initial Settings

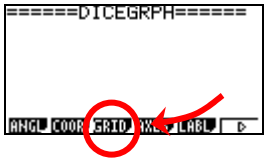
So now you know some of the types of things you can draw on the calculator. It is time to combine what you've learnt with what you know about programming. We are going to now write some graphic programs. Each one will be slightly more complex than the previous.

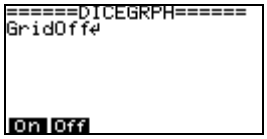
1. Enter PRGM mode from the main menu.

2. Create a new program by pressing New (F3) and name it DICEGRPH.

3. For any program, you need to set the scene for what you wish to do. In our case, we are going to prepare the calculator for drawing graphics. Recall the settings that we adjusted in "**LESSON 2 – Adjusting Peripheral Settings**". The regular features that we are interested in are the Axes, Grid, Label, ViewWindow and Cls. Enter the **SETUP** menu (**SHIFT MENU**). To have a clear screen, we need turn off the axes, grid and labels.

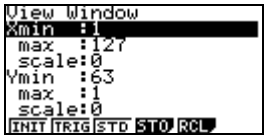
4. Enter the GRID (F3) menu and press OFF (F2), then press **EXE**. Some screenshots are provided below to help you.



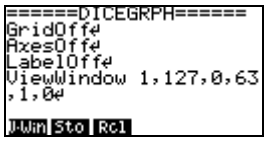


5. Press EXIT to get back a screen to the SETUP menu. Now enter AXES (F4), turn the setting OFF (F2) and press **EXE**. Now EXIT and do the same with the LABEL setting.

6. Next is the ViewWindow. Remember the settings looked like this:



We are going to insist on these numbers again but will write it into the program as one line. Press **V-Window** (**SHIFT** F3), then press V-Win (F1). Now enter the number shown in the figure above, in the same order and separated by a comma each time. Press **EXE** at the end.



7. The last part is to enter the Cls command. Ensure your cursor is on a new line below the V-Window.

8. Open the SKETCH menu and press Cls (F1).

That's it, the scene is set.

How it works
When a person uses your program, you will not know the settings of their calculator. It is crucial then to include settings like these at the beginning of your programs.

This collection of commands does exactly the same thing we performed in **LESSON 2 – Adjusting peripheral settings** and will be used in all the programs that follow.

LESSON 7 – Your First Cover Page

We are going to make a cover page for your SUMDICE program you wrote in the previous booklet. You don't need the SUMDICE program to do this lesson though.

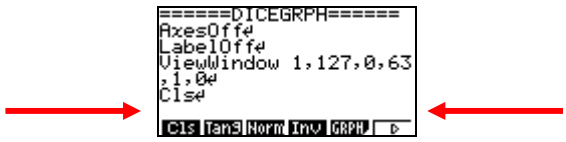
You'll be instructed how to do an action and explained what it means, but when there are similar actions required, you will just be told what is needed.

The graphic we are going to design looks like this.

So here we go.

1. If you are not already in the program DICEGRPH, then enter it now.

2. Place the cursor on the line below the V-Window command.



3. Now what we are going to do is construct a cube in 2D. This is going to comprise of a whole stack of F-Lines. In terms of what the coordinates should be, all it takes is some little common sense and a whole stack of trial and error. Press 'Des' (F6) until you see LINE (F2). Enter that menu and press F-Line (F2).

4. Below are the coordinates of the vertices of the cube. The edges are 20 pixels long.

For each line in the picture, we need to write an F-Line command. Now enter in the following lines.

```
ViewWindow 1,127,0,63
,1,0↵
Cls↵
F-Line 20,20,20,40↵
F-Line 20,40,40,40↵
F-Line 40,40,40,20↵
F-Line 40,20,20,20↵
F-Line 20,20,30,15↵
F-Line 30,15,50,15↵
F-Line 50,15,40,20↵
F-Line 50,15,50,35↵
F-Line 50,35,40,40↵
```

It seems like a lot of work and it is. Obviously this is not the most efficient method of drawing graphics because it requires brute force, but unfortunately there are some pictures that can only be done this way. Press EXIT and run the program.

5. Now on to the text. Once again, the positioning comes from re-jigging it many times. Insert your name in the section that says "YOUR NAME". This part will require simply inserting text into the picture, so further explanation will be needed. Enter the following lines after the group of F-Lines.

```
F-Line 50,15,50,35↵
F-Line 50,35,40,40↵
Text 20,70,"SUM DICE"
↵
Text 32,83,"BY"↵
Text 40,68,"YOUR NAME
"↵
```

6. That should do it! Exit the program and give it a try. You will most likely have to adjust the position of your name though.

Why Not Try...

1. Try and place a number in the face(s) of the die.
2. If you are really feeling brave, use the PxlOn command and draw some dots on the face of the die. Remember that the opposite sides of a die always add up to 7!

LESSON 8 – Using Loops to Draw Graphics

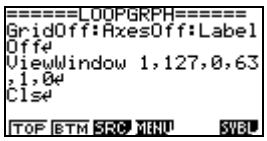
Many of the programs involve loops and if you are not entirely comfortable with them, then I recommend you look back to the first booklet and read over "**LESSON 13 – Introducing a For statement**".

When there is a pattern in a graphic, sometimes you can use a loop to draw the parts. This reduces both the size of the file in the calculator and the workload of the programmer. We are going to start with a basic loop to perform a basic graphic. The final product of this lesson will be like below:

1. Start a new program and call it LOOPGRPH.

2. We are going to include all those precursors that we included at the start of the last program. Notice that before we used a new line for every command. You can continue to use this but now you'll be shown something different.

   Instead of having each command on a new line, it is possible to use the one. Enter the **SETUP** menu and choose GRID (F3) then OFF (F2). Next, open the **PRGM** menu, then 'Des' (F6). You will see a colon symbol (:). This means the exact same thing as the return symbol (↵) but allows you to keep working on the same line. Go ahead and enter the rest of the commands shown below.

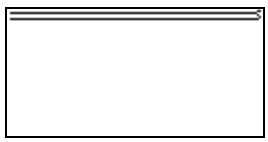


3. Remember that the window is set so there are 127 points across. We are going to use a loop to place a piece of text repeatedly across the window. This will be done using a For statement, because of the incremental properties of it. Enter the following line at the bottom of the program.

```
Cls↵
For 1→Z To 127↵
Text 1,Z,"S"↵
Next↵
```

   Since Z will increase by 1 each time it goes through the For statement, the text will slide across by 1.

4. Exit the program and run it to see what happens. It should look like the picture shown below.



   What is happening is that each S is being typed just right of the previous one. Obviously we need to adjust the horizontal spacing so there is enough room to see each letter.

5. Enter back into the LOOPGRPH program.

6. Place the cursor on the ↵ in the For 1→Z To 127 line, and set it **INS** (**SHIFT DEL**).

```
For 1→Z To 127↵
```
Here

7. Recall that for a For statement, the variable increments by 1. This is the default setting that can be changed. Enter the PRGM menu and press 'Des' (F6) until you see the For command. Notice that there is a command called Step (F4).

8. This Step command tells the calculator what the size of the increment should be. Just by trial-and-error, a nice number to use is 6, so the new line should look like:

```
For 1→Z To 127 Step 6
↵
```

9. Give it try and see how it looks.

10. Enter back into the LOOPGRPH program.

11. Now we are going to add the rest of the word "SUMDICE". We are going to arrange it so that each row is filled with the same letter, but a column will read SUMDICE. This is simply done by using the same For statement we've been using, but will now include all the other letters. Once again through trial and error, separating each row by 8 pixels works rather well.

```
For 1→Z To 127 Step 6
↵
Text 1,Z,"S"↵
Text 9,Z,"U"↵
Text 17,Z,"M"↵
Text 25,Z,"D"↵
Text 33,Z,"I"↵
Text 41,Z,"C"↵
Text 49,Z,"E"↵
Next↵
```

12. Exit the program and give it a run.

13. Enter back into LOOPGRPH. Now all we have to do is write your name in. Enter the last line of text, adjusting it as needed.

```
Text 49,Z,"E"↵
Text 57,30,"BY YOUR N
AME HERE"↵
```

14. Exit the program and give it a try.

How it works

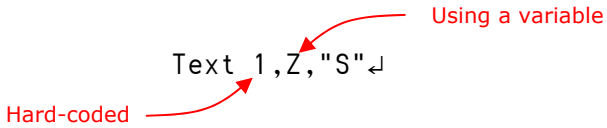
You've just finished your first graphic using a loop. This is really powerful because it allows you produce a lot of content with very little programming. The power of the For statement is that it runs through as many times as we need it to and we can also use the 'counting variable' (in this case, Z).

```
For 1→Z To 127 Step 6
↲
Text 1,Z,"S"↲
Text 9,Z,"U"↲
Text 17,Z,"M"↲
Text 25,Z,"D"↲
Text 33,Z,"I"↲
Text 41,Z,"C"↲
Text 49,Z,"E"↲
Next↲
```

For this program, you were introduced to the Step command. This is an optional setting you can use to adjust the default increment of the For statement.

Why Not Try…

1. To hone your skills and develop this idea of using a loop to draw graphics, why not try to rewrite LOOPGRPH so that the word "SUMDICE" is repeated horizontally.
2. **Challenge:** You'll notice that the vertical position of each letter has been 'hard-coded', meaning that it is a number, as opposed to a variable.



It is never a good idea to hard-code something where it can be easily avoided, because if you wish to change it, you have to change each individual number. On the other hand, if you use a variable like Z, you only have to change the initial value of Z.

See if you can translate the positions of the row in terms of Z. It is going to involve some mathematics and the rule will look something like this:

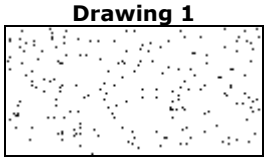
```
Text (number)xZ+(numb
er),Z,"S"↲
```

LESSON 9 – Using Ran# to Draw Graphics

All of the graphics we've created so far have been designed by us and the results are totally expected, but there are cool images you can make by adding randomness to your drawings. We are going to start off lightly with a basic random drawing and move onto more complex pictures as we build the skills.

You will need to recall the Ran# function on the calculator. As a quick recap, Ran# randomly generates numbers between 0 and 0.9999… You can multiply Ran# by other numbers and then take the integer value of it. For further explanation, refer to **Lesson 12 – Understanding the Ran# Function** in the first programming booklet I wrote.

Drawing 1



For this image, we are going to randomly turn pixels on.

1. Enter the PRGM mode on your calculator from your main menu.

2. Start a new program (F3) and call it RANPIXEL.

3. We are going to enter in our initial instructions like we do for all our drawings. Ensure they are the same as shown below.

4. The plan is to randomly generate two numbers for the *x* and *y* coordinates, then turn the pixel located at that point on. We will loop the program through 200 times.

   Firstly, enter a For statement from 1 to 200, using C as your variable.

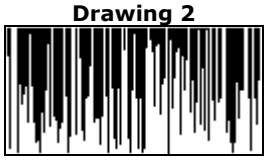
```
For 1→C To 200↵
```

5. Now for the random locations and the completion of the For statement.

```
For 1→C To 200↵
Int (127Ran#+1)→A↵      ← Random x-coordinate
Int (63Ran#+1)→B↵       ← Random y-coordinate
PxlOn B,A↵              ← Random pixel at (y, x)
Next↵
```

6. Exit RANPIXEL and see it run. It is sure to look different to **Drawing 1**. Well done! You finished your first random picture.

Drawing 2



This drawing will work along the same lines as RANPIXEL, but we will use F-Lines instead.

1. Start a new program called RANVLINE.

2. Put in the initial instructions now for the Axes, Grid, Label, V-Window and Cls.

3. The plan of attack is to draw 200 vertical lines. Each line will begin at the top of the screen, will be a random length, and will be randomly positioned across the screen. We are going to use a For loop to minimize the amount of code.

```
Cls↵
For 1→W To 200↵
Int (127Ran#+1)→A↵
Int (63Ran#+1)→B↵
F-Line A,1,A,B↵
Next↵
```

4. You're finished, can you believe it! Exit the program and give it a try.

How it works
Lets look at each section of the code one by one.

```
For 1→W To 200↵
```

This line begins our For statement, using the variable W to hold our variable. You can see it goes to 200 and since there is no Step command, the default increment is 1.

```
Int (127Ran#+1)→A↵
Int (63Ran#+1)→B↵
```

The first line randomly generates a number from 1 to 127 and stores the answer as variable A. We use this number as the horizontal position of our line (↔). The next line does the same but between 1 and 63, and is used for the vertical length of the line (↕).

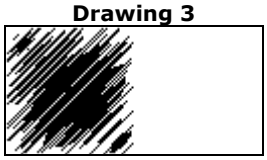
```
F-Line A,1,A,B↵
```

Here is the command that draws the line. Joining the points (A_y, 1_x) and (A_y, B_x). You will notice the *y*-coordinate is unchanged, which means it will it be a vertical line, but the line will be a distance of B pixels.

```
Next↵
```

This command repeatedly sends it back to the For command, until it is complete.

Drawing 3



This drawing will work along the same lines as RANVLINE, but we are going diagonal this time.

1. Start a new program called DIAGONAL.

2. Put in the initial instructions now for the Axes, Grid, Label, V-Window and Cls.

3. For this picture, we are going to randomly generate a coordinate, say (A_y, B_x) and then draw a line to (B_y, A_x). Obviously we cannot go greater than 63 because the line would finish below the screen. This creates a pretty interesting effect of only using half of the screen.

```
Cls↵
For 1→W To 200↵
Int (63Ran#+1)→A↵
Int (63Ran#+1)→B↵
F-Line A,B,B,A↵
Next↵
```

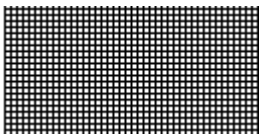
4. Give it a try!

Why Not Try…

1. Create a new program called RANHLINE. Make it a variation of RANVLINE by having the lines going horizontally this time.
2. Add to the program DIAGONAL by finishing the picture on the other side.

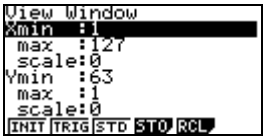
LESSON 10 – Using the Vert and Hztl commands

This drawing will is going to use two new commands that are very simple to understand. They are called Vert and Hztl and simply draw vertical and horizontal lines respectively.

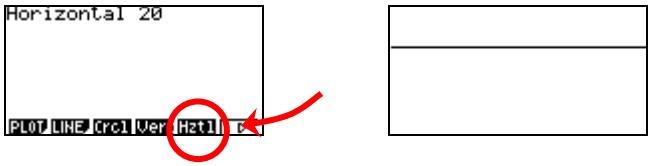


1. The in-built commands of Vert and Hztl are located in the **SKETCH** menu. So you can understand how these commands work, enter RUN mode from the main menu. Now open the **SKETCH** menu and press 'Des' (F6). There you will see the two commands, Vert and Hztl.

2. Adjust the V-WINDOW settings to the values we have used throughout this book.



3. Press EXIT. Now clear the screen using the Cls command. Go to the Hztl command in the **SKETCH** menu. Press Hztl (F5) and then enter 20. Press **EXE** and you will notice a horizontal line the full width of the screen 20 pixels from the top.



4. Now press **MENU** and enter PRGM mode. Start a new program called GRID.

5. Put in the initial instructions now for the Axes, Grid, Label, V-Window and Cls.

6. Enter in the code and it will be explained afterwards.

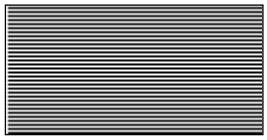
```
Cls↵
For 1→Z To 63 Step 3↵
Horizontal Z↵
Next↵
For 1→Z To 127 Step 3
↵
Vertical Z↵
Next↵
```

7. Give it a try!

How it works

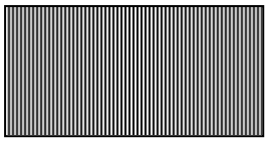
This program is broken into two parts – the horizontal lines and the vertical lines.

```
For 1→Z To 63 Step 3↵
Horizontal Z↵
Next↵
```



The section of code above draws all of the horizontal lines. Since the screen is 63 pixels high, our `For` statement goes from 1 to 63. The `Step 3` part is so there is a gap between adjacent lines. If we did not put this in, the screen would be completely filled in.

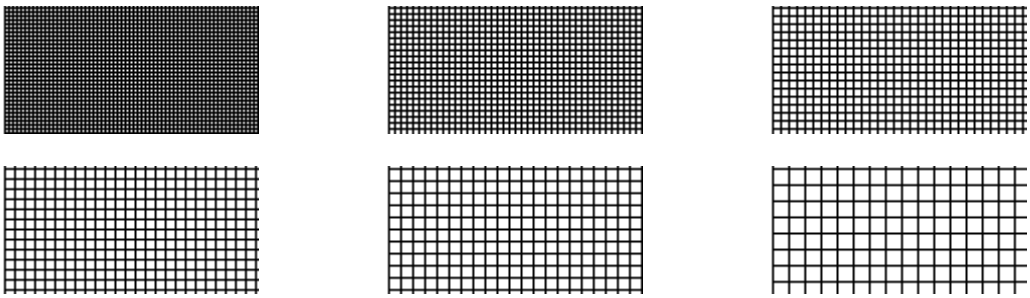
```
For 1→Z To 127 Step 3
↵
Vertical Z↵
Next↵
```



The section of code above draws all of the vertical lines is just like the horizontal portion of code, except it goes from 1 to 127 (since there are 127 pixels across) and uses the `Vert` command.

Why Not Try…

1. Adjust the `Step` values and see what the affect is.
2. Why not adjust the `GRID` program so that it asks the user the gap they want. You would use this as the `Step` value to create different spacings between the lines.



LESSON 11 – Saving a Picture

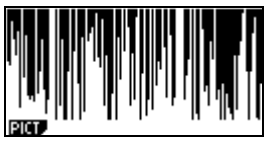
With all the good work we've done so far, it would be a shame if you couldn't somehow take a 'photo' of it. Fortunately there is a way you can record your good work.

1. Enter PRGM mode from the main menu.

2. Run the program RANVLINE.

3. When it has finished, press OPTN on the keypad.

4. You see a single menu appear on the bottom of the screen, PICT (F1). Enter this menu.



5. This opens another menu with two options, STO (F1) which stands for store and RCL (F2) which stands for recall. Press STO (F1).

6. You now have the option to save your masterpiece. You can save up to 6 pictures, memory allowing that is. Choose Pic1 (F1).

7. Press **AC/ON** button to clear your screen. You can be in RUN mode to view it again. Fortunately programs are executed in RUN mode anyway, so we are already in there.

8. We are looking for the PICT command again. It is in the OPTN menu like last time, but we have to press 'Des' (F6) a few times to find it.

9. When you see PICT (F2), enter it and then enter RCL (F2). The list of pictures comes up again. Press Pic1 (F1) to see you picture again.

LESSON 12 – Freehand drawing

This is a fantastic thing you can use to do freehand drawings. You will be using the commands we have used so far, plus some new ones. Drawing like this involves some patience but the pictures can look great. Since you will be drawing on the run, **DO NOT press EXIT, MENU, ARROWS or ON/AC**. If you do this, your picture will be lost forever. I suggest you regularly 'store your' picture using the method mentioned in the previous lesson.

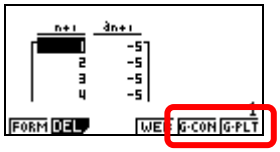
I would like to thank two of my Year 7 boys, *Matt French* and *Daniel Lee*, for telling me about this.

This lesson will be creating a basic picture. I apologise beforehand for my poor artistic talent!

1. From your main menu, enter RECUR mode. Don't worry, we won't be doing anything with recursion! Make sure your window settings are the values we have been using.

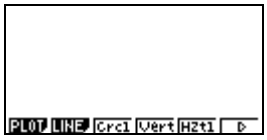
2. Enter the value of –5 in a_{n+1}. We picked this value because it is a number outside the window settings of our V-Window.

3. Press TABL (F6). You will see a table of values which is irrelevant to us. Press either G–CON (F5) or G–PLT (F6).



4. This produces a blank screen, which for us, is our blank canvas.

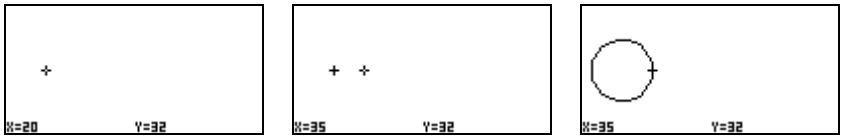
5. Our tools are in the **SKETCH** menu. Open that now. You will only see the Cls (F1) option and 'Des' (F6). Press 'Des' and you will see more options. Press it again and you will see PEN (F1) and Text (F2). These are all our drawing tools in this mode.



6. You are going to use the most common tools. Firstly, let's draw a circle. Move the menu around using 'Des' until you see Crcl (F3). Press F3 and you will see a little red cross appear in the middle of the screen. You can move this around using the arrows on your keypad.

If you have a fair distance to move it on the screen, you can hold down the arrow.

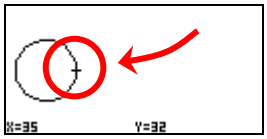
We are going to place the arrow where we want the centre of our circle to be. Place it some off to the left and press **EXE**. Now press the right arrow and you will see another cross appear. This is now measuring the radius of our circle. You could have moved it in any direction. When you have moved it several pixels to the right, press **EXE** again. You will see the circle being created.



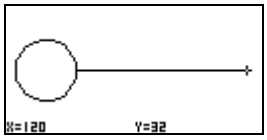
7. Enter the SKETCH menu.

8. Next we'll draw an F–Line. In the SKETCH menu, you will see LINE (F2). When you enter this menu, F–Line will appear. Select F–Line (F2). The cross appears again in the centre of the screen.

9. Move the cross to the edge of the circle. Press **EXE**.



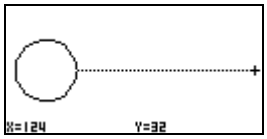
10. Move your cross to the right again. Like with the circle, another cross appears but with a line also appears. Drag it to the right edge.



11. It is a good practice to save regularly, so we'll do that now. Press OPTN, then PICT (F1), STO (F1) and Pic1 (F1). Well done.

12. The next part will take some patience. We are going to 'hash' the line we drew in step 10. In the SKETCH menu, you will see the menu PLOT (F1). Enter this and then Pl–off (F3). Obviously this turns pixels off. Move the cross to the start of the line. Press the **EXE**. Move the cursor away and you will see the pixel has been turned off. *Fortunately the PLOT options do not require you go back into the SKETCH menu repeatedly for successive uses*.

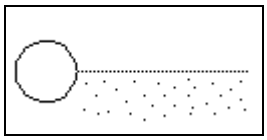
    Continue this for every second pixel in the line.



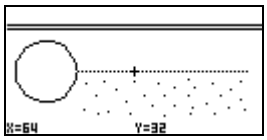
13. Save your work like you did in step 11.

14. Now we'll put some dots below the line. Enter the PLOT menu again and this time choose Plot (F1).

15. Move your cross below the line and press **EXE**. You'll see that the pixel has been turned on. Continue you this until you think there is enough.



16. Horizontal and vertical lines are easy to draw. Enter the SKETCH menu now and you will see Hrtl (F5). Press F5 and you will see a red horizontal line. This tells you exactly where the line would be if you pressed **EXE**. Move the cursor near the top and press **EXE**. Move it down two pixels and press **EXE** again. *Fortunately the Hrtl and Vrtl options do not require you go back into the SKETCH menu repeatedly for successive uses*.



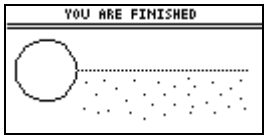
17. Save your work

18. We are going to place some text above the line. It is good to have a rough idea where it should go before you do anything, because it would be a tedious pain to turn each pixel off (using Pl-off) if it was in the wrong spot.

    Enter the SKETCH menu and then move to the Text (F2) option. Press Text and move the cursor to two pixels from the top edge of the screen and across about 35 pixels. The cross is the place of the of the very top left hand corner of the text.

THE TEXT STARTS FROM THE CROSS

    You can put the A-LOCK on if you do not wish to press ALPHA for each letter. Enter the text "YOU ARE FINISHED".



19. Congratulations, YOU ARE FINISHED! Save your work.

The Final Programs

Filename:DICEGRPH

```
GridOff↵
AxesOff↵
LabelOff↵
ViewWindow 1,127,0,63,1,0↵
Cls↵
F-Line 20,20,20,40↵
F-Line 20,40,40,40↵
F-Line 40,40,40,20↵
F-Line 40,20,20,20↵
F-Line 20,20,30,15↵
F-Line 30,15,50,15↵
F-Line 50,15,40,20↵
F-Line 50,15,50,35↵
F-Line 50,35,40,40↵
Text 20,70,"SUM DICE"↵
Text 32,83,"BY"↵
Text 40,68,"YOUR NAME"↵
```

Filename:LOOPGRPH

```
GridOff:AxesOff:LabelOff↵
ViewWindow 1,127,0,63,1,0↵
Cls↵
For 1→Z To 127 Step 6↵
Text 1,Z,"S"↵
Text 9,Z,"U"↵
Text 17,Z,"M"↵
Text 25,Z,"D"↵
Text 33,Z,"I"↵
Text 41,Z,"C"↵
Text 49,Z,"E"↵
Next↵
```

Filename:RANPIXEL

```
GridOff:AxesOff:LabelOff↵
ViewWindow 1,127,0,63,1,0↵
Cls↵
Int (127Ran#+1)→A↵
Int (63Ran#+1)→B↵
PxlOn B,A↵
Next↵
```

Filename:RANPIXEL

```
GridOff:AxesOff:LabelOff↵
ViewWindow 1,127,0,63,1,0↵
Cls↵
For 1→W To 200↵
Int (127Ran#+1)→A↵
Int (63Ran#+1)→B↵
F-Line A,1,A,B↵
Next↵
```

Filename:DIAGONAL

```
GridOff:AxesOff:LabelOff↵
ViewWindow 1,127,0,63,1,0↵
Cls↵
For 1→V To 200↵
Int (63Ran#+1)→A↵
Int (63Ran#+1)→B↵
F-Line A,B,B,A↵
Next↵
```

Filename:GRID

```
GridOff:AxesOff:LabelOff↵
ViewWindow 1,127,0,63,1,0↵
Cls↵
For 1→Z To 63 Step 3↵
Horizontal Z↵
Next↵
For 1→Z To 127 Step 3↵
Vertical Z↵
Next↵
```

Final Word

I would like to thank you very much for taking the time to read my books and for the kind emails I have received from people around the world. I hope the booklets have been useful to you and that they have helped you learn about programming.

If you have enjoyed programming, the best thing you could do now is to look for other things to do. There is a *Casio Web Ring* who has churned out an enormous quantity of material. You'll find them from a Google search. At the CasioEd Australia Education website, we have programs called ADDs that help students with mathematical concepts. You can freely submit any you have to be hosted there.

I recommend you download Casio's FA123 program from the CasioEd Australia Education website. With this you can write programs more easily, take screen shots and place them in documents (like I have), transfer data from lists and a few other things. It is completely free and simple to install.

Please feel comfortable emailing CasioEd with any questions and comments. Thanks again.

Warm regards,
Marty Schmude